CLAIMS

We claim:

1. A fast encoder for compressing input data into output compressed data, **comprising**:

at least one single-level direct subband transformer (**200**, **201**, ...), for receiving and transforming input data to produce transformation coefficients;

at least one encoding probability estimator (**260**, **261**, ...) coupled to appropriate said single-level direct subband transformer, for receiving the transformation coefficients and estimating the probabilities of symbols within the specified contexts to produce the probabilities of symbols within the specified contexts;

at least one entropy encoder (**280**, **281**, ...) coupled to appropriate said encoding probability estimator, for receiving and entropy encoding the transformation coefficients using the probabilities of symbols within the specified contexts to produce encoded data; and

an output compressed buffer (**32**) coupled to said entropy encoders, for receiving and substantially synchronizing the encoded data with said fast encoder to produce output compressed data,

whereby said fast encoder performs lossless compression.

2. The fast encoder of claim **1**, **further comprising**

at least one quantizer (**240**, **241**, ...) coupled to appropriate said single-level direct subband transformer, for receiving and quantizing the transformation coefficients to produce quantized transformation coefficients, **wherein**:

each said encoding probability estimator is coupled to appropriate said quantizer, for receiving the quantized transformation coefficients and estimating the probabilities of symbols within the specified contexts to produce the probabilities of symbols within the specified contexts; and

each said entropy encoder is coupled to appropriate said encoding probability estimator, for receiving and entropy encoding the quantized transformation coefficients using the probabilities of symbols within the specified contexts to produce encoded data,

whereby said fast encoder performs lossy compression.

3. The fast encoder of claim **1**, further comprising

at least one synchronization memory (**420**, **421**, ...) coupled to appropriate said entropy encoder, for receiving and substantially synchronizing the encoded data with a fast encoder to produce synchronized compressed data, **wherein**

said output compressed buffer is coupled to said synchronization memories, for receiving and buffering synchronized compressed data to produce the output compressed data.

4. The fast encoder of claim **1**, **further comprising**

at least one color space converter for converting an original input image to produce the input data.

5. The fast encoder of claim **1**, **wherein**:

first said single-level direct subband transformer is coupled to receive and transform the input data to produce transformation coefficients; and

each other said single-level direct subband transformer is coupled to receive and transform selected transformation coefficients to produce transformed transformation coefficients.

6. The fast encoder of claim **5**, **wherein** selected transformation coefficients are low-pass transformed for one-dimensional input data.

7. The fast encoder of claim **5**, **wherein** selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

8. The fast encoder of claim **1**, **wherein**

said single-level direct subband transformer **comprises**:

at least one direct filter for horizontal filtering; and

at least one direct filter for vertical filtering.

9. The fast encoder of claim **8**, **wherein** said direct filter for horizontal filtering is different from said direct filter for vertical filtering.

10. The fast encoder of claim **8, wherein**

at least one of said direct filter for horizontal filtering and said direct filter for vertical filtering **comprises**

at least one direct non-stationary filter.

11. The fast encoder of claim **1, wherein**

said single-level direct subband transformer **comprises**

at least one direct filter for filtering.

12. The fast encoder of claim **11, wherein**

said direct filter **comprises**

at least one direct non-stationary filter.

13. The fast encoder of claim **12, wherein**

said direct non-stationary filter **comprises**

a plurality of serially coupled direct non-stationary filter cells.

14. The fast encoder of claim **13, wherein**

said direct non-stationary filter cell **comprises**:

a filter device **(805)**;

a filter cell input x coupled to said filter device **(805)**;

a filter cell output y coupled to said filter device **(805)**;

a first switch **(800)** and a second switch **(801)** coupled to said filter device **(805)**, having a plurality of positions controlled by a clock input c; and

a clock input c coupled to control said first switch **(800)** and said second switch **(801)**, for providing a non-stationarity of said direct non-stationary filter cell.

15. The fast encoder of claim **14, wherein**:

said first switch **(800)** is in the first position for the horizontal filtering of each second pixel and in the second position for the horizontal filtering of other pixels; and

said second switch **(801)** is in the second position for the horizontal filtering of each second pixel and in the first position for the horizontal filtering of other pixels.

16. The fast encoder of claim **14, wherein**:

said first switch **(800)** is in the first position for the vertical filtering of each second line and in the second position for the vertical filtering of other lines; and

said second switch **(801)** is in the second position for the vertical filtering of each second line and in the first position for the vertical filtering of other lines.

17. The fast encoder of claim **14, wherein**

said direct non-stationary filter **further comprises**:

a first gain multiplier **(881)**;

a second gain multiplier **(882)**; and

a selection switch **(880)**, having a plurality of positions controlled by said clock input c, **wherein**

an output of said plurality of serially coupled direct non-stationary filter cells is coupled to an input of said first gain multiplier **(881)**, for multiplying said output with a first gain number to produce a first result;

an output of said plurality of serially coupled direct non-stationary filter cells is coupled to an input of said second gain multiplier **(882)**, for multiplying said output with a second gain number to produce a second result;

an output of said direct non-stationary filter is coupled to an output of said first gain multiplier **(881)** for said selection switch **(880)** in the first position; and

an output of said direct non-stationary filter is coupled to an output of said second gain multiplier **(882)** for said selection switch **(880)** in the second position.

18. The fast encoder of claim **14, wherein** said filter device **comprises**:

at least one delay element z^{-w} **(500, 501,..., 500+m-2)**;

a plurality of multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)**, $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)**; and

a plurality of adders **(700, 701, 702, 703,..., 700+m-4, 700+m-3, 700+m-2, 700+m-1)**, **wherein**:

an output of each even indexed said delay element z^{-w} **(500, 502,..., 500+m-4)** is coupled to an input of subsequent odd indexed said delay element z^{-w} **(501, 503,..., 500+m-3)**;

an output of each odd indexed said delay element z^{-w} **(501, 503,..., 500+m-3)** is coupled to an input of subsequent even indexed said delay element z^{-w} **(502, 504,..., 500+m-2)**;

the output of each even indexed said delay element z^{-w} **(500, 502,..., 500+m-2)** is coupled to an input of appropriate said multiplier $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)**;

outputs of all said multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)** are coupled to inputs of said adders **(701, 703,..., 700+m-3)**, for adding together all outputs of all said multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)** to produce a first result;

inputs of first said adder **(700)** are coupled to receive and add the first result with said filter cell input x;

an input of first said delay element z^{-w} **(500)** is coupled to said filter cell input x for said first switch **(800)** in the first position;

an input of first said delay element z^{-w} **(500)** is coupled to the output of first said adder **(700)** for said first switch **(800)** in the second position;

said filter cell input x and the output of each odd indexed said delay element z^{-w} **(501, 503,..., 500+m-3)** is coupled to an input of appropriate said multiplier $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)**;

outputs of all said multipliers $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)** are coupled to inputs of said adders **(702, 704, ..., 700+m-2)**, for adding together all outputs of all said multipliers $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)** to produce a second result;

inputs of last said adder **(700+m-1)** are coupled to receive and add the second result with the output of last said delay element z^{-w} **(500+m-2)**;

said filter cell output y is coupled to the output of last said delay element z^{-w} **(500+m-2)** for said second switch **(801)** in the first position; and

said filter cell output y is coupled to the output of last said adder **(700+m-1)** for said second switch **(801)** in the second position.

19. The fast encoder of claim **18, wherein**

at least one of said multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)**, $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2) comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

20. The fast encoder of claim **18, wherein** said filter device **further comprises**

a first function N_1 means **(802)** coupled to receive and transform the first result to produce a third result; and

a second function N_2 means **(803)** coupled to receive and transform the second result to produce a fourth result,

wherein:

inputs of first said adder **(700)** are coupled to receive and add the third result with said filter cell input x; and

inputs of last said adder **(700+m-1)** are coupled to receive and add the fourth result with the output of last said delay element z^{-w} **(500+m-2)**.

21. The fast encoder of claim **20, wherein**

at least one of said first function N_1 means **(802)** and said second function N_2 means **(803) comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

22. The fast encoder of claim **14, wherein** said filter device **comprises**:

a delay element z^{-w} **(1500)**;

a first multiplier **(1600)** and a second multiplier **(1601)**; and

a first adder **(1700)** and a second adder **(1701)**,

wherein:

an input of said first multiplier **(1600)** is coupled to said filter cell input x;

an input of said second multiplier **(1601)** is coupled to an output of said delay element z^{-w} **(1500)**;

inputs of said first adder **(1700)** are coupled to receive and add the output of said second multiplier **(1601)** with said filter cell input x;

an input of said delay element z^{-w} (**1500**) is coupled to said filter cell input x for said first switch (**800**) in the first position;

an input of said delay element z^{-w} (**1500**) is coupled to the output of said first adder (**1700**) for said first switch (**800**) in the second position;

inputs of said second adder (**1701**) are coupled to receive and add the output of said first multiplier (**1600**) with the output of said delay element z^{-w} (**1500**);

said filter cell output y is coupled to the output of said delay element z^{-w} (**1500**) for said second switch (**801**) in the first position; and

said filter cell output y is coupled to the output of said second adder (**1701**) for said second switch (**801**) in the second position.

23. The fast encoder of claim **22**, **wherein**

at least one of said first multiplier (**1600**) and said second multiplier (**1601**) **comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

24. The fast encoder of claim **23**, **wherein** said shifting means shifts right data from its input for two bit positions to produce data at its output.

25. The fast encoder of claim **23**, **wherein** said shifting means shifts right data from its input for one bit position to produce data at its output.

26. The fast encoder of claim **25**, **wherein**

inputs of said first adder (**1700**) are coupled to receive and subtract the output of said second multiplier (**1601**) from said filter cell input x; and

inputs of said second adder (**1701**) are coupled to receive and subtract the output of said first multiplier (**1600**) from the output of said delay element z^{-w} (**1500**).

27. The fast encoder of claim **14**, **wherein** said filter device **comprises**:

a first delay element z^{-w} (**1540**), a second delay element z^{-w} (**1541**) and a third delay element z^{-w} (**1542**);

a first multiplier (**1640**), a second multiplier (**1641**), a third multiplier (**1642**) and a fourth multiplier (**1643**); and

a first adder (**1740**), a second adder (**1741**), a third adder (**1742**) and a fourth adder (**1743**), **wherein**:

an output of said first delay element z^{-w} (**1540**) is coupled to an input of said second delay element z^{-w} (**1541**) and an input of said second multiplier (**1641**);

an output of said second delay element z^{-w} (**1541**) is coupled to an input of said third delay element z^{-w} (**1542**) and an input of said third multiplier (**1642**);

an input of said fourth multiplier (**1643**) is coupled to the output of said third delay element z^{-w} (**1542**);

inputs of said second adder (**1741**) are coupled to receive and add the output of said second multiplier (**1641**) with the output of said fourth multiplier (**1643**);

inputs of said first adder (**1740**) are coupled to receive and add the output of said second adder (**1741**) with said filter cell input x;

an input of said first delay element z^{-w} (**1540**) is coupled to said filter cell input x for said first switch (**800**) in the first position;

an input of said first delay element z^{-w} (**1540**) is coupled to the output of said first adder (**1740**) for said first switch (**800**) in the second position;

inputs of said third adder (**1742**) are coupled to receive and add the output of said third multiplier (**1642**) with the output of said first multiplier (**1640**);

inputs of said fourth adder (**1743**) are coupled to receive and add the output of said third adder (**1742**) with the output of said third delay element z^{-w} (**1542**);

said filter cell output y is coupled to the output of said third delay element z^{-w} (**1542**) for said second switch (**801**) in the first position; and

said filter cell output y is coupled to the output of said fourth adder (**1743**) for said second switch (**801**) in the second position.

28. The fast encoder of claim **27**, **wherein**

at least one of said first multiplier (**1640**), said second multiplier (**1641**), said third multiplier (**1642**) and said fourth multiplier (**1643**) **comprises**

a shifting means selected from a group **consisting of**:
shifters and shifted hardwired bit line connections.

29. The fast encoder of claim **28, wherein** said shifting means shifts right data from its input for four bit positions to produce data at its output.

30. The fast encoder of claim **29, wherein**:
inputs of said second adder (**1741**) are coupled to receive and subtract the output of said second multiplier (**1641**) from the output of said fourth multiplier (**1643**); and
inputs of said third adder (**1742**) are coupled to receive and subtract the output of said third multiplier (**1642**) from the output of said first multiplier (**1640**).

31. The fast encoder of claim **1, wherein**
said encoding probability estimator **comprises**
at least one adaptive histogram updating means, for updating an adaptive histogram.

32. The fast encoder of claim **31, wherein**
said adaptive histogram updating means **comprises**
a low-pass filter for filtering probabilities selected from a group **consisting of**:
probabilities of occurrences of a current symbol x; and
cumulative probabilities of occurrences of all symbols preceding the current symbol x.

33. The fast encoder of claim **32, wherein**
said adaptive histogram updating means **further comprises**
a dominant pole adapter for adapting a dominant pole of said low-pass filter.

34. The fast encoder of claim **33, wherein**
said dominant pole adapter **comprises**
a dominant pole divider for halving a value of the dominant pole in each adaptation cycle.

35. The fast encoder of claim **1, wherein**
said entropy encoder is a range encoder, **comprising**

a first multiplier for multiplying a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

36. The fast encoder of claim **35, wherein** said first multiplier **comprises**:

a first simplified multiplier for multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$; and

a first left shifter coupled to said first simplified multiplier, for shifting left the output of said first simplified multiplier for l bit positions.

37. The fast encoder of claim **35, wherein** said first multiplier **comprises**

a first left shifter for shifting left said number $Q(x)$ for l bit positions.

38. The fast encoder of claim **35, wherein** said first multiplier **comprises**:

a third left shifter **comprising**:

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1, and

means for shifting left said number $Q(x)$ for one bit position, when said small number V is any odd number higher or equal 3;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

39. The fast encoder of claim **35, wherein** said first multiplier **comprises**:

a third left shifter **comprising**:

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1,

means for shifting left said number $Q(x)$ for one bit position, when said small number V is equal 3, and

means for shifting left said number $Q(x)$ for two bit positions, when said small number V is any odd number higher or equal 5;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

40. The fast encoder of claim **1, wherein**

said entropy encoder is a range encoder, **comprising**

a first divider for dividing a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

41. The fast encoder of claim **40, wherein** said first divider **comprises**

a first right shifter for shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

42. The fast encoder of claim **1, wherein**

said encoding probability estimator **comprises**:

a transformation coefficient C splitter into a sign S and a magnitude M;

a magnitude-set index MS determinator coupled to said transformation coefficient C splitter, for determining the magnitude-set index MS using said magnitude M and a magnitude-set table;

a residual R determinator, coupled to said transformation coefficient C splitter, for determining a residual R using said magnitude M and said magnitude-set table.

43. The fast encoder of claim **42, wherein**

said magnitude-set index MS is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude M; and

said residual R is determined as the difference between said magnitude M and the lower coefficient limit, equal to a value of said magnitude M with all bits zeroed except the first nonzero bit of the highest significance and the first next bit of the lower significance in a binary representation of said magnitude M.

44. The fast encoder of claim **42, wherein**

said entropy encoder **comprises**

a residual R encoder coupled to said residual R determinator, for encoding the residual R using variable length coding (VLC).

45. The fast encoder of claim **42, wherein**

said encoding probability estimator **further comprises**

a context modeler of a transformation coefficient to be encoded, using already encoded transformation coefficients.

46. The fast encoder of claim **45, wherein**

said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

47. The fast encoder of claim **45, wherein**

said encoding probability estimator **further comprises**

a mean value $\overline{MS}$ determinator coupled to said context modeler, for determining $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already encoded transformation coefficients.

48. The fast encoder of claim **47, wherein**

said encoding probability estimator **further comprises**

a maximum mean value $\overline{MS}$ limiter coupled to said mean value $\overline{MS}$ determinator, for limiting a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

49. The fast encoder of claim **48, wherein**

said entropy encoder **comprises**

a magnitude range encoder coupled to said maximum mean value $\overline{MS}$ limiter, for encoding said magnitude-set index MS as a current symbol x, using an adaptive magnitude histogram $h[MC]$.

50. The fast encoder of claim **48, wherein**

said encoding probability estimator **further comprises**

adaptive magnitude histogram $h[MC]$ updating means coupled to said maximum mean value $\overline{MS}$ limiter, for an adaptive magnitude histogram $h[MC]$ updating using said magnitude-set index MS as a current symbol x.

51. The fast encoder of claim **45, wherein**

said encoding probability estimator **further comprises**

a ternary context TC determinator coupled to said transformation coefficient C splitter, for determining a ternary context TC as the ternary code of sign values S_i of already encoded transformation coefficients.

52. The fast encoder of claim **51, wherein**

said encoding probability estimator **further comprises**

a sign inverter coupled to said ternary context TC determinator, for inverting less probable said sign S using NEG table.

53. The fast encoder of claim **51, wherein**

said encoding probability estimator **further comprises**

a ternary context TC translator coupled to said ternary context TC determinator, for translating said ternary context TC into a sign context SC using CTX table.

54. The fast encoder of claim **53, wherein**

said entropy encoder **comprises**

a sign range encoder coupled to said ternary context TC translator, for encoding said sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

55. The fast encoder of claim **53, wherein**

said encoding probability estimator **further comprises**

an adaptive sign histogram $g[SC]$ updating means coupled to said ternary context TC translator, for an adaptive sign histogram $g[SC]$ updating using said sign S as a current symbol x.

56. A fast decoder for decompressing input compressed data into output data, **comprising**:

an input compressed buffer (**33**), for receiving and substantially synchronizing the input compressed data with said fast decoder to produce synchronized compressed data;

at least one entropy decoder (**290**, **291**, ...) coupled to said input compressed buffer and appropriate decoding probability estimator, for receiving and decoding the synchronized compressed data using the probabilities of symbols within the specified contexts to produce transformation coefficients;

at least one decoding probability estimator (**270**, **271**, ...) coupled to appropriate said entropy decoder, for receiving the transformation coefficients and estimating the probabilities of symbols to produce the probabilities of symbols within the specified contexts; and

at least one single-level inverse subband transformer (**210**, **211**, ...) coupled to appropriate said entropy decoder, for receiving and transforming the transformation coefficients to produce the output data,

whereby said fast decoder performs lossless decompression.

57. The fast decoder of claim **56**, **further comprising**

at least one dequantizer (**250**, **251**, ...) coupled to appropriate said entropy decoder, for receiving and dequantizing the transformation coefficients to produce dequantized transformation coefficients, **wherein**

each said single-level inverse subband transformer is coupled to appropriate said dequantizer, for receiving and transforming the dequantized transformation coefficients to produce the output data,

whereby said fast decoder performs lossy decompression.

58. The fast decoder of claim **56**, **further comprising**

at least one synchronization memory (**430**, **431**, ...) coupled to said input compressed buffer, for receiving and substantially synchronizing buffered compressed data with said fast decoder to produce the synchronized compressed data, **wherein**:

said input compressed buffer is coupled to receive and buffer the input compressed data to produce the buffered compressed data; and

each said entropy decoder is coupled to appropriate said synchronization memory and appropriate said decoding probability estimator, for receiving and decoding the synchronized compressed data using the probabilities of symbols within the specified contexts to produce transformation coefficients.

59. The fast decoder of claim **56, further comprising**

at least one color space converter for converting the output data to produce converted output data.

60. The fast decoder of claim **56, wherein**:

last said single-level inverse subband transformer is coupled to receive and transform transformation coefficients to produce the output data; and

each other said single-level inverse subband transformer is coupled to receive and transform transformation coefficients to produce selected transformation coefficients.

61. The fast decoder of claim **60, wherein** said selected transformation coefficients are low-pass transformed for one-dimensional output data.

62. The fast decoder of claim **60, wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional output data.

63. The fast decoder of claim **56, wherein**

said single-level inverse subband transformer **comprises**:

at least one inverse filter for horizontal filtering; and

at least one inverse filter for vertical filtering.

64. The fast decoder of claim **63, wherein** said inverse filter for horizontal filtering is different from said inverse filter for vertical filtering.

65. The fast decoder of claim **63, wherein**

at least one of said inverse filter for horizontal filtering and said inverse filter for vertical filtering **comprises**

at least one inverse non-stationary filter.

66. The fast decoder of claim **56, wherein**

said single-level inverse subband transformer **comprises**

at least one inverse filter for filtering.

67. The fast decoder of claim **66, wherein**

said inverse filter **comprises**

at least one inverse non-stationary filter.

68. The fast decoder of claim **67, wherein**

said inverse non-stationary filter **comprises**

a plurality of serially coupled inverse non-stationary filter cells.

69. The fast decoder of claim **68, wherein**

said inverse non-stationary filter cell **comprises**:

a filter device **(805)**;

a filter cell input x coupled to said filter device **(805)**;

a filter cell output y coupled to said filter device **(805)**;

a first switch **(800)** and a second switch **(801)** coupled to said filter device **(805)**, having a plurality of positions controlled by a clock input c; and

a clock input c coupled to control said first switch **(800)** and said second switch **(801)**, for providing a non-stationarity of said direct non-stationary filter cell.

70. The fast decoder of claim **69, wherein**:

said first switch **(800)** is in the second position for the horizontal filtering of each second pixel and in the first position for the horizontal filtering of other pixels; and

said second switch **(801)** is in the first position for the horizontal filtering of each second pixel and in the second position for the horizontal filtering of other pixels.

71. The fast decoder of claim **69, wherein**

said first switch **(800)** is in the second position for the vertical filtering of each second line and in the first position for the vertical filtering of other lines; and

said second switch **(801)** is in the first position for the vertical filtering of each second line and in the second position for the vertical filtering of other lines.

72. The fast decoder of claim **69, wherein**

said direct non-stationary filter **further comprises:**

a first gain multiplier **(891)**;

a second gain multiplier **(892)**; and

a selection switch **(890)**, having a plurality of positions controlled by said clock input c,

wherein:

an input of said first gain multiplier **(891)** is coupled to an input of said inverse non-stationary filter, for multiplying an input sample with a reciprocal value of a first gain number to produce a first result;

an input of said second gain multiplier **(892)** is coupled to an input of said inverse non-stationary filter, for multiplying an input sample with a reciprocal value of a second gain number to produce a second result;

an input of said plurality of serially coupled inverse non-stationary filter cells is coupled to an output of said first gain multiplier **(891)**, for said selection switch **(890)** in the second position; and

an input of said plurality of serially coupled inverse non-stationary filter cells is coupled to an output of said second gain multiplier **(892)**, for said selection switch **(890)** in the first position.

73. The fast decoder of claim **69, wherein** said filter device **comprises:**

at least one delay element z^{-w} **(500, 501,..., 500+m-2)**;

a plurality of multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)**, $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)**; and

a plurality of adders **(700, 701, 702, 703,..., 700+m-4, 700+m-3, 700+m-2, 700+m-1)**,

wherein:

an output of each even indexed said delay element z^{-w} **(500, 502,..., 500+m-4)** is coupled to an input of subsequent odd indexed said delay element z^{-w} **(501, 503,..., 500+m-3)**;

an output of each odd indexed said delay element z^{-w} **(501, 503,..., 500+m-3)** is coupled to an input of subsequent even indexed said delay element z^{-w} **(502, 504,..., 500+m-2)**;

the output of each even indexed said delay element z^{-w} **(500, 502,..., 500+m-2)** is coupled to an input of appropriate said multiplier $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)**;

outputs of all said multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)** are coupled to inputs of said adders **(701, 703,..., 700+m-3)**, for adding together all outputs of all said multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)** to produce a first result;

inputs of first said adder **(700)** are coupled to receive and add the first result with said filter cell input x;

an input of first said delay element z^{-w} **(500)** is coupled to said filter cell input x for said first switch **(800)** in the first position;

an input of first said delay element z^{-w} **(500)** is coupled to the output of first said adder **(700)** for said first switch **(800)** in the second position;

said filter cell input x and the output of each odd indexed said delay element z^{-w} **(501, 503,..., 500+m-3)** is coupled to an input of appropriate said multiplier $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)**;

outputs of all said multipliers $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)** are coupled to inputs of said adders **(702, 704, ..., 700+m-2)**, for adding together all outputs of all said multipliers $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2)** to produce a second result;

inputs of last said adder **(700+m-1)** are coupled to receive and add the second result with the output of last said delay element z^{-w} **(500+m-2)**;

said filter cell output y is coupled to the output of last said delay element z^{-w} **(500+m-2)** for said second switch **(801)** in the first position; and

said filter cell output y is coupled to the output of last said adder **(700+m-1)** for said second switch **(801)** in the second position.

74. The fast decoder of claim **73**, wherein

at least one of said multipliers $K_1[0]$ **(601)**, $K_1[1]$ **(603)**,..., $K_1[k-1]$ **(600+m-1)**, $K_2[k-1]$ **(600)**, $K_2[k-2]$ **(602)**,..., $K_2[0]$ **(600+m-2) comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

75. The fast decoder of claim **73**, **wherein**

said inverse non-stationary filter cell **further comprises**

a first function N_1 means **(802)** coupled to receive and transform the first result to produce a third result; and

a second function N_2 means **(803)** coupled to receive and transform the second result to produce a fourth result,

wherein:

inputs of first said adder **(700)** are coupled to receive and add the third result with said filter cell input x; and

inputs of last said adder **(700+m-1)** are coupled to receive and add the fourth result with the output of last said delay element z^{-w} **(500+m-2)**.

76. The fast decoder of claim **75**, **wherein**

at least one of said first function N_1 means **(802)** and said second function N_2 means **(803) comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

77. The fast decoder of claim **69**, **wherein** said filter device **comprises**:

a delay element z^{-w} **(1510)**;

a first multiplier **(1610)** and a second multiplier **(1611)**; and

a first adder **(1710)** and a second adder **(1711)**,

wherein:

an input of said first multiplier **(1610)** is coupled to said filter cell input x;

an input of said second multiplier (**1611**) is coupled to an output of said delay element z^{-w} (**1510**);

inputs of said first adder (**1710**) are coupled to receive and add the output of said second multiplier (**1611**) with said filter cell input x;

an input of said delay element z^{-w} (**1510**) is coupled to said filter cell input x, for said first switch (**800**) in the first position;

an input of said delay element z^{-w} (**1510**) is coupled to the output of said first adder (**1710**), for said first switch (**800**) in the second position;

inputs of said second adder (**1711**) are coupled to receive and add the output of said first multiplier (**1610**) with the output of said delay element z^{-w} (**1510**);

said filter cell output y is coupled to the output of said delay element z^{-w} (**1510**), for said second switch (**801**) in the first position; and

said filter cell output y is coupled to the output of said second adder (**1711**), for said second switch (**801**) in the second position.

78. The fast decoder of claim **77**, **wherein**

at least one of said first multiplier (**1610**) and said second multiplier (**1611**) **comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

79. The fast decoder of claim **78**, **wherein** said shifting means shifts right data from its input for one bit position to produce data at its output.

80. The fast decoder of claim **78**, **wherein** said shifting means shifts right data from its input for two bit positions to produce data at its output.

81. The fast decoder of claim **80**, **wherein**

inputs of said first adder (**1710**) are coupled to receive and subtract the output of said second multiplier (**1611**) from said filter cell input x; and

inputs of said second adder (**1711**) are coupled to receive and subtract the output of said first multiplier (**1610**) from the output of said delay element z^{-w} (**1510**).

82. The fast decoder of claim **69**, **wherein** said filter device **comprises**:

a first delay element z^{-w} (**1550**), a second delay element z^{-w} (**1551**) and a third delay element z^{-w} (**1552**);

a first multiplier (**1650**), a second multiplier (**1651**), a third multiplier (**1652**) and a fourth multiplier (**1653**); and

a first adder (**1750**), a second adder (**1751**), a third adder (**1752**) and a fourth adder (**1753**), **wherein**:

an output of said first delay element z^{-w} (**1550**) is coupled to an input of said second delay element z^{-w} (**1551**) and an input of said second multiplier (**1651**);

an output of said second delay element z^{-w} (**1551**) is coupled to an input of said third delay element z^{-w} (**1552**) and an input of said third multiplier (**1652**);

an input of said fourth multiplier (**1653**) is coupled to the output of said third delay element z^{-w} (**1552**);

inputs of said second adder (**1751**) are coupled to receive and add the output of said fourth multiplier (**1653**) with the output of said second multiplier (**1651**);

inputs of said first adder (**1750**) are coupled to receive and add the output of said second adder (**1751**) with said filter cell input x;

an input of said first delay element z^{-w} (**1550**) is coupled to said filter cell input x, for said first switch (**800**) in the first position;

an input of said first delay element z^{-w} (**1550**) is coupled to the output of said first adder (**1750**), for said first switch (**800**) in the second position;

inputs of said third adder (**1752**) are coupled to receive and add the output of said first multiplier (**1650**) with the output of said third multiplier (**1652**);

inputs of said fourth adder (**1753**) are coupled to receive and add the output of said third adder (**1752**) with the output of said third delay element z^{-w} (**1552**);

said filter cell output y is coupled to the output of said third delay element z^{-w} (**1552**), for said second switch (**801**) in the first position; and

said filter cell output y is coupled to the output of said fourth adder (**1753**), for said second switch (**801**) in the second position.

83. The fast decoder of claim **82**, **wherein**

at least one of said first multiplier (**1650**), said second multiplier (**1651**), said third multiplier (**1652**) and said fourth multiplier (**1653**) **comprises**

a shifting means selected from a group **consisting of**:

shifters and shifted hardwired bit line connections.

84. The fast decoder of claim **83**, **wherein** said shifting means shifts right data from its input for four bit positions to produce data at its output.

85. The fast decoder of claim **84**, **wherein**:

inputs of said second adder (**1751**) are coupled to receive and subtract the output of said fourth multiplier (**1653**) from the output of said second multiplier (**1651**); and

inputs of said third adder (**1752**) are coupled to receive and subtract the output of said first multiplier (**1650**) from the output of said third multiplier (**1652**).

86. The fast decoder of claim **56**, **wherein**

said decoding probability estimator **comprises**

at least one adaptive histogram updating means, for updating an adaptive histogram.

87. The fast decoder of claim **86**, **wherein**

said adaptive histogram updating means **comprises**

a low-pass filter for filtering probabilities selected from a group **consisting of**:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

88. The fast decoder of claim **87**, **wherein**

said adaptive histogram updating means **further comprises**

a dominant pole adapter for adapting a dominant pole of said low-pass filter.

89. The fast decoder of claim **88**, **wherein**

said dominant pole adapter **comprises**

a dominant pole divider for halving a value of the dominant pole in each adaptation cycle.

90. The fast decoder of claim **56, wherein**

said entropy decoder is a range decoder, **comprising**

a first multiplier for multiplying a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

91. The fast decoder of claim **90, wherein** said first multiplier **comprises**:

a first simplified multiplier for multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$; and

a first left shifter coupled to said first simplified multiplier, for shifting left the output of said first simplified multiplier for l bit positions.

92. The fast decoder of claim **90, wherein** said first multiplier **comprises**

a first left shifter for shifting left said number $Q(x)$ for l bit positions.

93. The fast decoder of claim **90, wherein** said first multiplier **comprises**:

a third left shifter **comprising**:

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1, and

means for shifting left said number $Q(x)$ for one bit position, when said small number V is any odd number higher or equal 3;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

94. The fast decoder of claim **90, wherein** said first multiplier **comprises**:

a third left shifter **comprising:**

means for zeroing its output, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1,

means for shifting left said number $Q(x)$ for one bit position, when said small number V is equal 3, and

means for shifting left said number $Q(x)$ for two bit positions, when said small number V is any odd number higher or equal 5;

a first adder coupled to said third left shifter, for adding said number $Q(x)$ with the output of said third left shifter; and

a first left shifter coupled to said first adder, for shifting left the output of said first adder for l bit positions.

95. The fast decoder of claim **56, wherein**

said entropy decoder is a range decoder, **comprising**

a first divider for dividing a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

96. The fast decoder of claim **95, wherein** said first divider **comprises**

a first right shifter for shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

97. The fast decoder of claim **56, wherein**

said entropy decoder is a range decoder, **comprising**

a second divider for dividing a bottom range limit B with a prescaled range r, to produce a range correction $t = \lfloor B/r \rfloor$.

98. The fast decoder of claim **97, wherein** said second divider **comprises:**

a second simplified divider for dividing said bottom range limit B with a small number $V = \lfloor r \cdot 2^{-l} \rfloor$; and

a second right shifter coupled to said second simplified divider, for shifting right the output of said second simplified divider for l bit positions.

99. The fast decoder of claim **97**, **wherein** said second divider **comprises:**

a third multiplier for multiplying said bottom range limit B with a first predefined number, dependent on a small number $V = \lfloor r \cdot 2^{-l} \rfloor$; and

a second right shifter coupled to said third multiplier, for shifting right the output of said third multiplier for a sum of l and a second predefined number of bit positions, dependent on said small number V.

100. The fast decoder of claim **56**, **wherein**

said decoding probability estimator **comprises**

a transformation coefficient C builder for reconstructing transformation coefficient C using a magnitude-set index MS, a sign S and a residual R.

101. The fast decoder of claim **100**, **wherein**

said entropy decoder **comprises**

a residual R decoder, for decoding said residual R using inverse variable length coding (INVVLC).

102. The fast decoder of claim **100**, **wherein**

said decoding probability estimator **further comprises**

a context modeler of a transformation coefficient to be decoded, using already decoded transformation coefficients.

103. The fast decoder of claim **102**, **wherein**

said already decoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be decoded.

104. The fast decoder of claim **102**, **wherein**

said decoding probability estimator **further comprises**

a mean value $\overline{MS}$ determinator coupled to said context modeler, for determining $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already decoded transformation coefficients.

105. The fast decoder of claim **104, wherein**

said decoding probability estimator **further comprises**

a maximum mean value $\overline{MS}$ limiter coupled to said mean value $\overline{MS}$ determinator, for limiting a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

106. The fast decoder of claim **105, wherein**

said entropy decoder **comprises**

a magnitude range decoder coupled to said maximum mean value $\overline{MS}$ limiter, for decoding a magnitude-set index MS as a current symbol x, using an adaptive magnitude histogram $h[MC]$.

107. The fast decoder of claim **106, wherein**

said decoding probability estimator **further comprises**

adaptive magnitude histogram $h[MC]$ updating means coupled to said magnitude range decoder, for an adaptive magnitude histogram $h[MC]$ updating using decoded said magnitude-set index MS as a current symbol x.

108. The fast decoder of claim **102, wherein**

said decoding probability estimator **further comprises**

a ternary context TC determinator coupled to said context modeler, for determining a ternary context TC as the ternary code of sign values S_i of already decoded transformation coefficients.

109. The fast decoder of claim **108, wherein**

said decoding probability estimator **further comprises**

a ternary context TC translator coupled to said ternary context TC determinator, for translating ternary context TC into a sign context SC using CTX table.

110. The fast decoder of claim **109, wherein**

said entropy decoder **comprises**

a sign range decoder coupled to said ternary context TC translator, for decoding a sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

111. The fast decoder of claim **110, wherein**

said decoding probability estimator **further comprises**

adaptive sign histogram $g[SC]$ updating means coupled to said sign range decoder, for an adaptive sign histogram $g[SC]$ updating using decoded said sign S as a current symbol x.

112. The fast decoder of claim **110, wherein**

said decoding probability estimator **further comprises**

a sign inverter coupled to said sign range decoder, for inverting less probable said sign S using NEG table.

113. A method for fast encoding input data into output compressed data, **comprising:**

direct subband transforming the input data to produce transformation coefficients;

estimating the probabilities of symbols within the specified contexts using the transformation coefficients to produce the probabilities of symbols within the specified contexts;

entropy encoding transformation coefficients using the probabilities of symbols within the specified contexts to produce encoded data; and

substantially synchronizing the encoded data to produce output compressed data,

whereby said method for fast encoding performs lossless compression.

114. The method for fast encoding of claim **113, further comprising**

quantizing transformation coefficients to produce quantized transformation coefficients, **wherein:**

said estimating the probabilities of symbols within the specified contexts is performed using the quantized transformation coefficients to produce the probabilities of symbols within the specified contexts; and

said entropy encoding quantization transformation coefficients is performed using the probabilities of symbols within the specified contexts to produce encoded data,

whereby said method for fast encoding performs lossy compression.

115. The method for fast encoding of claim **113, further comprising**

substantially synchronizing the encoded data to produce synchronized compressed data, **wherein**

buffering the synchronized compressed data is performed to produce output compressed data.

116. The method for fast encoding of claim **113, further comprising**

color space converting original input data to produce input data.

117. The method for fast encoding of claim **113, wherein**

said direct subband transforming **comprises**:

(a) direct subband transforming the input data to produce transformation coefficients;

(b) direct subband transforming selected transformation coefficients to produce transformed transformation coefficients; and

(c) repeating step (b) finite number of times.

118. The method for fast encoding of claim **117, wherein** said selected transformation coefficients are low-pass transformed for one-dimensional input data.

119. The method for fast encoding of claim **117, wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

120. The method for fast encoding of claim **113, wherein**

said direct subband transforming **comprises**:

at least one horizontal direct filtering; and

at least one vertical direct filtering.

121. The method for fast encoding of claim **120, wherein** said horizontal direct filtering is different from said vertical direct filtering.

122. The method for fast encoding of claim **120, wherein** at least one of said horizontal direct filtering and said vertical direct filtering **comprises** at least one direct non-stationary filtering.

123. The method for fast encoding of claim **113, wherein** said direct subband transforming **comprises** at least one direct filtering.

124. The method for fast encoding of claim **123, wherein** said direct filtering **comprises** at least one direct non-stationary filtering.

125. The method for fast encoding of claim **124, wherein** said direct non-stationary filtering **comprises** a plurality of successive direct non-stationary cell filtering steps.

126. The method for fast encoding of claim **125, wherein** said direct non-stationary cell filtering **comprises**: filtering using first direct transfer function in the first cycle; and filtering using second direct transfer function in the second cycle.

127. The method for fast encoding of claim **126, wherein** said first cycle is active during horizontal filtering of each second pixel; and said second cycle is active during horizontal filtering of other pixels.

128. The method for fast encoding of claim **126, wherein** said first cycle is active during vertical filtering of each second line; and said second cycle is active during vertical filtering of other lines.

129. The method for fast encoding of claim **126, further comprising:**

first multiplying a result of said plurality of successive direct non-stationary cell filtering steps with a first gain number to produce a first result;

second multiplying a result of said plurality of successive direct non-stationary cell filtering steps with a second gain number to produce a second result;

selecting the first result in each first cycle to produce an output sample; and

selecting the second result in each second cycle to produce the output sample.

130. The method for fast encoding of claim **126, wherein**

said direct non-stationary cell filtering **further comprises**:

delaying an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiplying each even indexed delayed result with appropriate first filter coefficient selected from a group of first filter coefficients to produce first results;

adding together all first results to produce a third result;

adding the third result with the input sample to produce a fifth result;

delaying the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiplying the input sample and each odd indexed delayed result with appropriate second filter coefficient selected from a group of second filter coefficients to produce second results;

adding together all second results to produce a fourth result;

adding the fourth result with last delayed result to produce a sixth result;

outputting the sixth result in each first cycle; and

outputting the last delayed result in each second cycle.

131. The method for fast encoding of claim **130, wherein**

at least one said multiplying **comprises**

an operation selected from a group **consisting of**:

shifting and bit remapping.

132. The method for fast encoding of claim **130, wherein**

said direct non-stationary cell filtering **further comprises**:

transforming the third result by first function N_1 to produce a seventh result;

transforming the fourth result by first function N_2 to produce an eight result;

adding the seventh result with the input sample to produce a fifth result; and
adding the eight result with last delayed result to produce a sixth result.

133. The method for fast encoding of claim **132, wherein** at least one said transforming **comprises** an operation selected from a group **consisting of**: shifting and bit remapping.

134. The method for fast encoding of claim **126, wherein** said direct non-stationary cell filtering **further comprises**:
delaying an input sample for *w* samples to produce a delayed result in each first cycle;
second multiplying the delayed result with a second filter coefficient to produce a second result;
first adding the second result with the input sample to produce a fourth result;
delaying the fourth result for *w* samples to produce the delayed result in each second cycle;
first multiplying the input sample with a first filter coefficient to produce a first result;
second adding the first result with the delayed result to produce a third result;
outputting the third result in each first cycle; and
outputting the delayed result in each second cycle.

135. The method for fast encoding of claim **134, wherein** at least one of said first multiplying and said second multiplying **comprises** an operation selected from a group **consisting of**: shifting and bit remapping.

136. The method for fast encoding of claim **135, wherein** said operation **comprises** shifting right for two bit positions.

137. The method for fast encoding of claim **135, wherein** said operation **comprises** shifting right for one bit position.

138. The method for fast encoding of claim **137, wherein**

said first adding **comprises** subtracting the second result from the input sample to produce a fourth result; and

said second adding **comprises** subtracting the first result from the delayed result to produce a third result.

139. The method for fast encoding of claim **126, wherein**

said direct non-stationary cell filtering **further comprises**:

delaying the input sample for w samples to produce a first delayed result in each first cycle;

delaying the first delayed result for w samples to produce a second delayed result;

delaying the second delayed result for w samples to produce a third delayed result;

first multiplying the input sample with a first filter coefficient to produce a first result;

second multiplying the first delayed result with a second filter coefficient to produce a second result;

third multiplying the second delayed result with a third filter coefficient to produce a third result;

fourth multiplying the third delayed result with a fourth filter coefficient to produce a fourth result;

second adding the second result with the fourth result to produce a sixth result;

third adding the third result with the first result to produce a fifth result;

first adding the sixth result with the input sample to produce an eight result;

delaying the eight result for w samples to produce the first delayed result in each second cycle;

fourth adding the fifth result with the third delayed result to produce a seventh result;

outputting the seventh result in each first cycle; and

outputting the third delayed result in each second cycle.

140. The method for fast encoding of claim **139, wherein**

at least one of said first multiplying, said second multiplying, said third multiplying and said fourth multiplying **comprises**

an operation selected from a group **consisting of**:

shifting and bit remapping.

141. The method for fast encoding of claim **140, wherein** said operation **comprises** shifting right for four bit positions.

142. The method for fast encoding of claim **141, wherein**:

said second adding **comprises** subtracting the second result from the fourth result to produce a sixth result; and

said third adding **comprises** subtracting the third result from the first result to produce a fifth result.

143. The method for fast encoding of claim **113, wherein** said estimating the probabilities of symbols within the specified contexts **comprises** updating adaptive histograms.

144. The method for fast encoding of claim **143, wherein** said updating adaptive histograms **comprises** low-pass filtering probabilities selected from a group **consisting of**:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

145. The method for fast encoding of claim **144, wherein** said updating adaptive histograms **further comprises** adapting a dominant pole during said low-pass filtering.

146. The method for fast encoding of claim **145, wherein** said adapting a dominant pole **comprises** halving value of the dominant pole in each adaptation cycle.

147. The method for fast encoding of claim **113, wherein** said entropy encoding is range encoding, **comprising** first multiplying a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

148. The method for fast encoding of claim **147, wherein** said first multiplying **comprises:**

simplified multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shifting left the first result for l bit positions.

149. The method for fast encoding of claim **147, wherein** said first multiplying **comprises**

shifting left said number $Q(x)$ for l bit positions.

150. The method for fast encoding of claim **147, wherein** said first multiplying **comprises:**

zeroing a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for l bit positions.

151. The method for fast encoding of claim **147, wherein** said first multiplying **comprises:**

zeroing a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shifting left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for l bit positions.

152. The method for fast encoding of claim **113, wherein**

said entropy encoding is range encoding, **comprising**

first dividing a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

153. The method for fast encoding of claim **152, wherein** said first dividing **comprises** shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

154. The method for fast encoding of claim **113, wherein**

said estimating the probabilities of symbols within the specified contexts **comprises**

splitting a transformation coefficient C into a sign S and a magnitude M;

determining a magnitude-set index MS using said magnitude M and a magnitude-set table; and

determining a residual R using said magnitude M and said magnitude-set table.

155. The method for fast encoding of claim **154, wherein**

said magnitude-set index MS is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude M; and

said residual R is determined as the difference between said magnitude M and the lower coefficient limit, equal to a value of said magnitude M with all bits zeroed except the first nonzero bit of the highest significance and the first next bit of the lower significance in a binary representation of said magnitude M.

156. The method for fast encoding of claim **154, wherein**

said entropy encoding **comprises**

encoding a residual R using variable length coding (VLC).

157. The method for fast encoding of claim **154, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

context modeling a transformation coefficient to be encoded, using already encoded transformation coefficients.

158. The method for fast encoding of claim **157, wherein**

said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

159. The method for fast encoding of claim **157, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

determining a mean value $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already encoded transformation coefficients.

160. The method for fast encoding of claim **159, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

limiting a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

161. The method for fast encoding of claim **160, wherein**

said entropy encoding **comprises**

range encoding said magnitude-set index MS as a current symbol x, using an adaptive magnitude histogram $h[MC]$.

162. The method for fast encoding of claim **160, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

updating of an adaptive magnitude histogram $h[MC]$ using said magnitude-set index MS as a current symbol x,.

163. The method for fast encoding of claim **162, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

determining a ternary context TC as the ternary code of sign values S_i of said already encoded transformation coefficients.

164. The method for fast encoding of claim **163, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

inverting less probable said sign S using NEG table.

165. The method for fast encoding of claim **163, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

translating said ternary context TC into a sign context SC using CTX table.

166. The method for fast encoding of claim **165, wherein**

said entropy encoder **comprises**

range encoding said sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

167. The method for fast encoding of claim **165, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

updating of an adaptive sign histogram $g[SC]$ using said sign S as a current symbol x.

168. A method for fast decoding of input compressed data into output data, **comprising:**

substantially synchronizing the input compressed data to produce synchronized compressed data;

entropy decoding the synchronized compressed data using the probabilities of symbols within the specified contexts to produce transformation coefficients;

estimating the probabilities of symbols within the specified contexts using the transformation coefficients to produce the probabilities of symbols within the specified contexts; and

inverse subband transforming the transformation coefficients to produce the output data, **whereby** said method for fast decoding performs lossless decompression.

169. The method for fast decoding of claim **168, further comprising**

dequantizing transformation coefficients to produce dequantized transformation coefficients, **wherein**

said inverse subband transforming the dequantized transformation coefficients is performed to produce the output data,

whereby said method for fast decoding performs lossy decompression.

170. The method for fast decoding of claim **168, further comprising**

buffering the input compressed data to produce buffered compressed data, **wherein**

said substantially synchronizing the buffered compressed data is performed to produce synchronized compressed data.

171. The method for fast decoding of claim **168, further comprising**

color space converting the output data to produce converted output data.

172. The method for fast decoding of claim **168, wherein**

said inverse subband transforming **comprises:**

(a) inverse subband transforming transformation coefficients to produce selected transformation coefficients;

(b) repeating step (a) finite number of times; and

(c) inverse subband transforming transformation coefficients to produce the output data.

173. The method for fast decoding of claim **172, wherein** said selected transformation coefficients are low-pass transformed for one-dimensional output data.

174. The method for fast decoding of claim **172, wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional output data.

175. The method for fast decoding of claim **168, wherein**

said inverse subband transforming **comprises:**

at least one horizontal inverse filtering; and
at least one vertical inverse filtering.

176. The method for fast decoding of claim **175**, **wherein** said horizontal inverse filtering is different from said vertical inverse filtering.

177. The method for fast decoding of claim **175, wherein**
at least one of said horizontal inverse filtering and said vertical inverse filtering **comprises**
at least one inverse non-stationary filtering.

178. The method for fast decoding of claim **168, wherein**
said inverse subband transforming **comprises**
at least one inverse filtering.

179. The method for fast decoding of claim **178, wherein**
said inverse filtering **comprises**
at least one inverse non-stationary filtering.

180. The method for fast decoding of claim **179, wherein**
said inverse non-stationary filtering **comprises**
a plurality of successive inverse non-stationary cell filtering steps.

181. The method for fast decoding of claim **180, wherein**
said inverse non-stationary cell filtering **comprises**:
filtering using first inverse transfer function in the first cycle; and
filtering using second inverse transfer function in the second cycle.

182. The method for fast decoding of claim **181, wherein**
said second cycle is active during horizontal filtering of each second pixel; and
said first cycle is active during horizontal filtering of other pixels.

183. The method for fast decoding of claim **181, wherein**

said second cycle is active during vertical filtering of each second line; and

said first cycle is active during vertical filtering of other lines.

184. The method for fast decoding of claim **181, wherein**

said inverse non-stationary cell filtering **further comprising**:

first multiplying an input with a reciprocal value of a first gain number to produce a first result;

second multiplying an input with a reciprocal value of the second gain number to produce a second result;

selecting first result in each second cycle to produce an input sample for said plurality of successive inverse non-stationary cell filtering steps; and

selecting second result in each first cycle to produce the input sample for said plurality of successive inverse non-stationary cell filtering steps.

185. The method for fast decoding of claim **181, wherein**

said inverse non-stationary cell filtering **further comprises**:

delaying an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiplying each even indexed delayed result with appropriate first filter coefficient selected from a group of first filter coefficients to produce first results;

adding together all first results to produce a third result;

adding the third result with the input sample to produce a fifth result;

delaying the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiplying the input sample and each odd indexed delayed result with appropriate second filter coefficient selected from the group of second filter coefficients to produce second results;

adding together all second results to produce a fourth result;

adding the fourth result with last delayed result to produce a sixth result;

outputting the sixth result in each first cycle; and

outputting the last delayed result in each second cycle.

186. The method for fast decoding of claim **185, wherein** at least one said multiplying **comprises** an operation selected from a group **consisting of**: shifting and bit remapping.

187. The method for fast decoding of claim **185, wherein** said inverse non-stationary cell filtering **further comprises**:
transforming the third result by first function N_1 to produce a seventh result;
transforming the fourth result by first function N_2 to produce an eight result;
adding the seventh result with the input sample to produce a fifth result; and
adding the eight result with last delayed result to produce a sixth result.

188. The method for fast decoding of claim **187, wherein** at least one said transforming **comprises** an operation selected from a group **consisting of**: shifting and bit remapping.

189. The method for fast decoding of claim **181, wherein** said inverse non-stationary cell filtering **further comprises**:
delaying an input sample for w samples to produce a delayed result in each first cycle;
second multiplying the delayed result with a second filter coefficient to produce a second result;
first adding the second result with the input sample to produce a fourth result;
delaying the fourth result for w samples to produce the delayed result in each second cycle;
first multiplying the input sample with a first filter coefficient to produce a first result;
second adding the first result with the delayed result to produce a third result;
outputting the third result in each first cycle; and
outputting the delayed result in each second cycle.

190. The method for fast decoding of claim **189, wherein** at least one of said first multiplying and said second multiplying **comprises**

an operation selected from a group **consisting of**: shifting and bit remapping.

191. The method for fast decoding of claim **190, wherein** said operation **comprises** shifting right for one bit position.

192. The method for fast decoding of claim **190, wherein** said operation **comprises** shifting right for two bit positions.

193. The method for fast decoding of claim **192, wherein**:

said first adding **comprises** subtracting the second result from the input sample to produce a fourth result; and

said second adding **comprises** subtracting the first result from the delayed result to produce a third result.

194. The method for fast decoding of claim **181, wherein**

said inverse non-stationary cell filtering **further comprises**:

delaying the input sample for w samples to produce a first delayed result in each first cycle;

delaying the first delayed result for w samples to produce a second delayed result;

delaying the second delayed result for w samples to produce a third delayed result;

first multiplying the input sample with a first filter coefficient to produce a first result;

second multiplying the first delayed result with a second filter coefficient to produce a second result;

third multiplying the second delayed result with a third filter coefficient to produce a third result;

fourth multiplying the third delayed result with a fourth filter coefficient to produce a fourth result;

second adding the fourth result with the second result to produce a sixth result;

third adding the first result with the third result to produce a fifth result;

first adding the sixth result with the input sample to produce an eight result;

delaying the eight result for w samples to produce the first delayed result in each second cycle;

fourth adding the fifth result with the third delayed result to produce a seventh result;
outputting the seventh result in each first cycle; and
outputting the third delayed result in each second cycle.

195. The method for fast decoding of claim **194, wherein**
at least one of said first multiplying, said second multiplying, said third multiplying and said fourth multiplying **comprises**
an operation selected from a group **consisting of:**
shifting and bit remapping.

196. The method for fast decoding of claim **195, wherein** said operation **comprises**
shifting right for four bit positions.

197. The method for fast decoding of claim **196, wherein:**
said second adding **comprises** subtracting the fourth result from the second result to produce a sixth result; and
said third adding **comprises** subtracting the first result from the third result to produce a fifth result.

198. The method for fast decoding of claim **168, wherein**
said estimating the probabilities of symbols within the specified contexts **comprises**
updating adaptive histograms.

199. The method for fast decoding of claim **198, wherein**
said updating adaptive histograms **comprises**
low-pass filtering probabilities selected from a group **consisting of:**
probabilities of occurrences of a current symbol x; and
cumulative probabilities of occurrences of all symbols preceding said current symbol x.

200. The method for fast decoding of claim **199, wherein**
said updating adaptive histograms **further comprises**
adapting a dominant pole during said low-pass filtering.

201. The method for fast decoding of claim **200, wherein** said adapting a dominant pole **comprises** halving value of the dominant pole in each adaptation cycle.

202. The method for fast decoding of claim **168, wherein** said entropy decoding is range decoding, **comprising** first multiplying a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

203. The method for fast decoding of claim **202, wherein** said first multiplying **comprises**:

simplified multiplying a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shifting left the first result for l bit positions.

204. The method for fast decoding of claim **202, wherein** said first multiplying **comprises** shifting left said number $Q(x)$ for l bit positions.

205. The method for fast decoding of claim **202, wherein** said first multiplying **comprises**:

zeroing a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for l bit positions.

206. The method for fast decoding of claim **202, wherein** said first multiplying **comprises**:

zeroing a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shifting left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shifting left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

adding said number $Q(x)$ with the first result to produce a second result; and

shifting left the second result for l bit positions.

207. The method for fast decoding of claim **168, wherein**

said entropy decoding is range decoding, **comprising**

first dividing a range R with a number $Total$ of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

208. The method for fast decoding of claim **207, wherein** said first dividing **comprises**

shifting right said range R for $w_3 = \log_2(Total)$ bit positions.

209. The method for fast decoding of claim **168, wherein**

said entropy decoding is range decoding, **comprising**

second dividing a bottom range limit B with a prescaled range r, to produce a range correction $t = \lfloor B/r \rfloor$.

210. The method for fast decoding of claim **209, wherein** said second dividing **comprises**:

simplified dividing said bottom range limit B with a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shifting right the first result for l bit positions.

211. The method for fast decoding of claim **209, wherein** said second dividing **comprises**:

multiplying said bottom range limit B with a first predefined number, dependent on a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shifting right the first result for a sum of l and a second predefined number of bit positions, dependent on said small number V.

212. The method for fast decoding of claim **168, wherein**

said estimating the probabilities of symbols within the specified contexts **comprises**

reconstructing transformation coefficient C using a magnitude-set index MS, a sign S and a residual R.

213. The method for fast decoding of claim **212, wherein**

said entropy decoding **comprises**

decoding said residual R using inverse variable length coding (INVVLC).

214. The method for fast decoding of claim **212, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

context modeling of a transformation coefficient to be decoded, using already decoded transformation coefficients.

215. The method for fast decoding of claim **214, wherein**

said already decoded transformation coefficients are located north-east, north, north-west and west from the transformation coefficient to be decoded.

216. The method for fast decoding of claim **214, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

determining a mean value $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already decoded transformation coefficients.

217. The method for fast decoding of claim **216, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

limiting a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

218. The method for fast decoding of claim **217, wherein**

said entropy decoding **comprises**

range decoding a magnitude-set index MS as a current symbol x, using an adaptive magnitude histogram $h[MC]$.

219. The fast decoding of claim **218, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

updating of an adaptive magnitude histogram $h[MC]$ using decoded said magnitude-set index MS as a current symbol x.

220. The method for fast decoding of claim **214, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

determining a ternary context TC as the ternary code of sign values S_i of said already decoded transformation coefficients.

221. The method for fast decoding of claim **220, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

translating said ternary context TC into a sign context SC using CTX table.

222. The method for fast decoding of claim **221, wherein**

said entropy decoding **comprises**

range decoding a sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

223. The method for fast decoding of claim **222, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

updating of an adaptive sign histogram $g[SC]$ using decoded said sign S as a current symbol x.

224. The method for fast decoding of claim **222, wherein**

said estimating the probabilities of symbols within the specified contexts **further comprises**

inverting less probable decoded said sign *S* using NEG table.

225. An article of manufacture for fast encoding of input data into output compressed data **comprising** a storage medium with a machine readable code **causing the machine to**:

direct subband transform the input data to produce transformation coefficients;

estimate the probabilities of symbols within the specified contexts using the transformation coefficients to produce the probabilities of symbols within the specified contexts;

entropy encode transformation coefficients using the probabilities of symbols within the specified contexts to produce encoded data; and

substantially synchronize the encoded data to produce the output compressed data,

whereby machine performs lossless compression.

226. The article of manufacture for fast encoding of claim **225, further causing the machine to**

quantize the transformation coefficients to produce quantized transformation coefficients, **wherein**:

said step that estimates the probabilities of symbols within the specified contexts is performed using the quantized transformation coefficients to produce the probabilities of symbols within the specified contexts; and

said entropy encode quantized transformation coefficients is performed using the probabilities of symbols within the specified contexts to produce encoded data,

whereby machine performs lossy compression.

227. The article of manufacture for fast encoding of claim **225, further causing the machine to**

substantially synchronize the encoded data to produce synchronized compressed data, **wherein**

buffering the synchronized compressed data is performed to produce the output compressed data.

228. The article of manufacture for fast encoding of claim **225, further causing the machine to** color space convert original input data to produce input data.

229. The article of manufacture for fast encoding of claim **225, wherein**

said step that direct subband transforms **causes the machine to**:

(a) direct subband transform the input data to produce transformation coefficients;

(b) direct subband transform selected transformation coefficients to produce transformed transformation coefficients; and

(c) repeat step (b) finite number of times.

230. The article of manufacture for fast encoding of claim **229, wherein** said selected transformation coefficients are low-pass transformed for one-dimensional input data.

231. The article of manufacture for fast encoding of claim **229, wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

232. The article of manufacture for fast encoding of claim **225, wherein**

said step that direct subband transforms **causes the machine to**:

direct filter horizontally; and

direct filter vertically.

233. The article of manufacture for fast encoding of claim **232, wherein** said step that direct filters horizontally is different from said step that direct filters vertically.

234. The article of manufacture for fast encoding of claim **232, wherein**

at least one of said step that direct filters horizontally and said step that direct filters vertically **causes the machine to**

direct non-stationary filter.

235. The article of manufacture for fast encoding of claim **225, wherein** said step that direct subband transforms **causes the machine to** direct filter.

236. The article of manufacture for fast encoding of claim **235, wherein** said step that direct filters **causes the machine to** direct non-stationary filter.

237. The article of manufacture for fast encoding of claim **236, wherein** said step that direct non-stationary filters **causes the machine to** perform a plurality of successive direct non-stationary cell filterings.

238. The article of manufacture for fast encoding of claim **237, wherein** said step that direct non-stationary cell filters **causes the machine to**:
filter using first direct transfer function in the first cycle; and
filter using second direct transfer function in the second cycle.

239. The article of manufacture for fast encoding of claim **238, wherein**
said first cycle is active during horizontal filtering of each second pixel; and
said second cycle is active during horizontal filtering of other pixels.

240. The article of manufacture for fast encoding of claim **238, wherein**
said first cycle is active during vertical filtering of each second line; and
said second cycle is active during vertical filtering of other lines.

241. The article of manufacture for fast encoding of claim **238, wherein** said step that direct non-stationary filters **further causes the machine to**:

multiply a result of a plurality of steps that successive direct non-stationary cell filter with a first gain number to produce a first result;

multiply a result of a plurality of steps that successive direct non-stationary cell filter with a second gain number to produce a second result;

select the first result in each first cycle to produce an output sample; and

select the second result in each second cycle to produce the output sample.

242. The article of manufacture for fast encoding of claim **238, wherein**

said step that direct non-stationary cell filters **further causes the machine to**:

delay an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiply each even indexed delayed result with appropriate first filter coefficient selected from a group of first filter coefficients to produce first results;

add together all first results to produce a third result;

add the third result with the input sample to produce a fifth result;

delay the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiply the input sample and each odd indexed delayed result with appropriate second filter coefficient selected from a group of second filter coefficients to produce second results;

add together all second results to produce a fourth result;

add the fourth result with last delayed result to produce a sixth result;

output the sixth result in each first cycle; and

output the last delayed result in each second cycle.

243. The article of manufacture for fast encoding of claim **242, wherein**

at least one said step that multiplies **causes the machine to**

perform an operation selected from a group **consisting of**:

shift and bit remap.

244. The article of manufacture for fast encoding of claim **242, wherein**

said step that direct non-stationary filters **further causes the machine to**:

transform the third result by first function N_1 to produce a seventh result;

transform the fourth result by first function N_2 to produce an eight result;

add the seventh result with the input sample to produce a fifth result; and

add the eight result with last delayed result to produce a sixth result.

245. The article of manufacture for fast encoding of claim **244, wherein** at least one said step that transforms **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

246. The article of manufacture for fast encoding of claim **238, wherein** said step that direct non-stationary cell filters **further causes the machine to**:
delay an input sample for w samples to produce a delayed result in each first cycle;
multiply the delayed result with a second filter coefficient to produce a second result;
add the second result with the input sample to produce a fourth result;
delay the fourth result for w samples to produce the delayed result in each second cycle;
multiply the input sample with a first filter coefficient to produce a first result;
add the first result with the delayed result to produce a third result;
output the third result in each first cycle; and
output the delayed result in each second cycle.

247. The article of manufacture for fast encoding of claim **246, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

248. The article of manufacture for fast encoding of claim **247, wherein** said operation **causes the machine to** shift right data for two bit positions.

249. The article of manufacture for fast encoding of claim **247, wherein** said operation **causes the machine to** shift right data for one bit position.

250. The article of manufacture for fast encoding of claim **249, wherein** said step that adds the second result **causes the machine to** subtract the second result from the input sample to produce a fourth result; and

said step that adds the first result **causes the machine to** subtract the first result from the delayed result to produce a third result.

251. The article of manufacture for fast encoding of claim **238, wherein**

said step that direct non-stationary cell filters **further causes the machine to**:

delay the input sample for *w* samples to produce a first delayed result in each first cycle;

delay the first delayed result for *w* samples to produce a second delayed result;

delay the second delayed result for *w* samples to produce a third delayed result;

multiply the input sample with a first filter coefficient to produce a first result;

multiply the first delayed result with a second filter coefficient to produce a second result;

multiply the second delayed result with a third filter coefficient to produce a third result;

multiply the third delayed result with a fourth filter coefficient to produce a fourth result;

add the second result with the fourth result to produce a sixth result;

add the third result with the first result to produce a fifth result;

add the sixth result with the input sample to produce an eight result;

delay the eight result for *w* samples to produce the first delayed result in each second cycle;

add the fifth result with the third delayed result to produce a seventh result;

output the seventh result in each first cycle; and

output the third delayed result in each second cycle.

252. The article of manufacture for fast encoding of claim **251, wherein**

at least one said step that multiplies **causes the machine to**

perform an operation selected from a group **consisting of**:

shift and bit remap.

253. The article of manufacture for fast encoding of claim **252, wherein**

said operation **causes the machine to**

shift right data for four bit positions.

254. The article of manufacture for fast encoding of claim **253, wherein**:

said step that adds the second result **causes the machine to** subtract the second result from the fourth result to produce a sixth result; and

said step that adds the third result **causes the machine to** subtract the third result from the first result to produce a fifth result.

255. The article of manufacture for fast encoding of claim **225, wherein**

said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**

update adaptive histograms.

256. The article of manufacture for fast encoding of claim **255, wherein**

said step that updates adaptive histograms **causes the machine to**

low-pass filter probabilities selected from a group **consisting of:**

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

257. The article of manufacture for fast encoding of claim **256, wherein**

said step that updates adaptive histograms **further causes the machine to**

adapt a dominant pole during said low-pass filtering.

258. The article of manufacture for fast encoding of claim **257, wherein**

said step that adapts a dominant pole **causes the machine to**

halve value of the dominant pole in each adaptation cycle.

259. The article of manufacture for fast encoding of claim **225, wherein**

said step that entropy encodes is a step that range encode, **causing the machine to**

multiply a prescaled range r with a number $Q(x)$ selected from a group **consisting of:**

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

260. The article of manufacture for fast encoding of claim **259, wherein**

said step that multiplies **causes the machine to:**

simplified multiply a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shift left the first result for l bit positions.

261. The article of manufacture for fast encoding of claim **259, wherein**

said step that multiplies **causes the machine to**

shift left said number $Q(x)$ for l bit positions.

262. The article of manufacture for fast encoding of claim **259, wherein**

said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

263. The article of manufacture for fast encoding of claim **259, wherein**

said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shift left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

264. The article of manufacture for fast encoding of claim **225, wherein**

said step that entropy encodes is a step that range encode, **causing the machine to**:

divide a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

265. The article of manufacture for fast encoding of claim **264, wherein**

said step that divides **causes the machine to**

shift right range R for $w_3 = \log_2(Total)$ bit positions.

266. The article of manufacture for fast encoding of claim **225, wherein**

said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**:

split a transformation coefficient C into a sign S and a magnitude M;

determine a magnitude-set index MS using said magnitude M and a magnitude-set table; and

determine a residual R using said magnitude M and said magnitude-set table.

267. The article of manufacture for fast encoding of claim **266, wherein**

said magnitude-set index MS is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude M; and

said residual R is determined as the difference between said magnitude M and the lower coefficient limit, equal to a value of said magnitude M with all bits zeroed except the first nonzero bit of the highest significance and the first next bit of the lower significance in a binary representation of said magnitude M.

268. The article of manufacture for fast encoding of claim **266, wherein**

said step that entropy encodes **causes the machine to**

encode a residual R using variable length coding (VLC).

269. The article of manufacture for fast encoding of claim **266, wherein**

said estimates the probabilities of symbols within the specified contexts **further causes the machine to**

context model a transformation coefficient to be encoded, using already encoded transformation coefficients.

270. The article of manufacture for fast encoding of claim **269, wherein**

said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

271. The article of manufacture for fast encoding of claim **269, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a mean value $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already encoded transformation coefficients.

272. The article of manufacture for fast encoding of claim **271, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

limit a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

273. The article of manufacture for fast encoding of claim **272, wherein**

said step that entropy encodes **causes the machine to**

range encode said magnitude-set index MS as a current symbol x, using said adaptive magnitude histogram $h[MC]$.

274. The article of manufacture for fast encoding of claim **272, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update an adaptive magnitude histogram $h[MC]$ using said magnitude-set index MS as a current symbol x.

275. The article of manufacture for fast encoding of claim **269, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a ternary context TC as the ternary code of sign values S_i of said already encoded transformation coefficients.

276. The article of manufacture for fast encoding of claim **275, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

invert less probable said sign S using NEG table.

277. The article of manufacture for fast encoding of claim **275, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

translate said ternary context TC into a sign context SC using CTX table.

278. The article of manufacture for fast encoding of claim **277, wherein**

said step that entropy encodes **causes the machine to**

range encode said sign S as a current symbol x, using said adaptive sign histogram $g[SC]$.

279. The article of manufacture for fast encoding of claim **277, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update of an adaptive sign histogram $g[SC]$ using said sign S as a current symbol x.

280. An article of manufacture for fast decoding of input compressed data into output data **comprising** a storage medium with a machine readable code **causing the machine to:**

substantially synchronize the input compressed data to produce synchronized compressed data;

entropy decode the synchronized compressed data using the probabilities of symbols within the specified contexts to produce transformation coefficients;

estimate the probabilities of symbols within the specified contexts using the transformation coefficients to produce the probabilities of symbols within the specified contexts; and

inverse subband transform the transformation coefficients to produce the output data,

whereby machine performs lossless decompression.

281. The article of manufacture for fast decoding of claim **280, further causing the machine to** dequantize transformation coefficients to produce dequantized transformation coefficients,

wherein

inverse subband transforming the dequantized transformation coefficients is performed to produce the output data,

whereby machine performs lossy decompression.

282. The article of manufacture for fast decoding of claim **280, further causing the machine to** buffer the input compressed data to produce buffered compressed data,

wherein

substantially synchronizing the buffered compressed data is performed to produce the synchronized compressed data.

283. The article of manufacture for fast decoding of claim **280, further causing the machine to** color space convert the output data to produce converted output data.

284. The article of manufacture for fast decoding of claim **280, wherein**

said step that inverse subband transforms **causes the machine to**:

(a) inverse subband transform transformation coefficients to produce selected transformation coefficients;

(b) repeat step (a) finite number of times; and

(c) inverse subband transform transformation coefficients to produce the output data.

285. The article of manufacture for fast decoding of claim **284, wherein** said selected transformation coefficients are low-pass transformed for one-dimensional input data.

286. The article of manufacture for fast decoding of claim **284, wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

287. The article of manufacture for fast decoding of claim **280, wherein**

said step that inverse subband transforms **causes the machine to**:

inverse filter horizontally; and

inverse filter vertically.

288. The article of manufacture for fast decoding of claim **287, wherein** said step that inverse filters horizontally is different from said step that inverse filters vertically.

289. The article of manufacture for fast decoding of claim **287, wherein**

at least one of said step that inverse filters horizontally and said step that inverse filters vertically **causes the machine to**

inverse non-stationary filter.

290. The article of manufacture for fast decoding of claim **280, wherein**

said step that inverse subband transforms **causes the machine to**

inverse filter.

291. The article of manufacture for fast decoding of claim **290, wherein**

said step that inverse filters **causes the machine to**

inverse non-stationary filter.

292. The article of manufacture for fast decoding of claim **291, wherein**

said step that inverse non-stationary filters **causes the machine to**

perform a plurality of successive inverse non-stationary cell filterings.

293. The article of manufacture for fast decoding of claim **292, wherein**

said step that inverse non-stationary cell filters **causes the machine to**:

filter using first direct transfer function in the first cycle; and

filter using second direct transfer function in the second cycle.

294. The article of manufacture for fast decoding of claim **293, wherein**

said second cycle is active during horizontal filtering of each second pixel; and

said first cycle is active during horizontal filtering of other pixels.

295. The article of manufacture for fast decoding of claim **293, wherein**
said second cycle is active during vertical filtering of each second line; and
said first cycle is active during vertical filtering of other lines.

296. The article of manufacture for fast decoding of claim **293, wherein**
said step that inverse non-stationary filters **further causes the machine to**:
multiply an input with a reciprocal value of a first gain number to produce a first result;
multiply an input with a reciprocal value of a second gain number to produce a second result;
select the first result in each second cycle to produce an input sample for a plurality of steps that successive inverse non-stationary cell filter; and
select the second result in each first cycle to produce an input sample for a plurality of steps that successive inverse non-stationary cell filter.

297. The article of manufacture for fast decoding of claim **293, wherein**
said step that inverse non-stationary cell filters **further causes the machine to**:
delay an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;
multiply each even indexed delayed result with appropriate first filter coefficient selected from a group of first filter coefficients to produce first results;
add together all first results to produce a third result;
add the third result with the input sample to produce a fifth result;
delay the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;
multiply the input sample and each odd indexed delayed result with appropriate second filter coefficient selected from a group of second filter coefficients to produce second results;
add together all second results to produce a fourth result;
add the fourth result with last delayed result to produce a sixth result;
output the sixth result in each first cycle; and
output the last delayed result in each second cycle.

298. The article of manufacture for fast decoding of claim **297, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

299. The article of manufacture for fast decoding of claim **297, wherein** said step that inverse non-stationary cell filters **further causes the machine to** transform the third result by a first function N_1 to produce a seventh result; transform the fourth result by a first function N_2 to produce an eight result; add the seventh result with the input sample to produce a fifth result; and add the eight result with last delayed result to produce a sixth result.

300. The article of manufacture for fast decoding of claim **299, wherein** at least one said step that transforms **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

301. The article of manufacture for fast decoding of claim **293, wherein** said step that inverse non-stationary cell filters **further causes the machine to**: delay an input sample for w samples to produce a delayed result in each first cycle; multiply the delayed result with a second filter coefficient to produce a second result; add the second result with the input sample to produce a fourth result; delay the fourth result for w samples to produce the delayed result in each second cycle; multiply the input sample with a first filter coefficient to produce a first result; add the first result with the delayed result to produce a third result; output the third result in each first cycle; and output the delayed result in each second cycle.

302. The article of manufacture for fast decoding of claim **301, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

303. The article of manufacture for fast decoding of claim **302, wherein**
said operation **causes the machine to**
shift right for one bit position.

304. The article of manufacture for fast decoding of claim **302, wherein**
said operation **causes the machine to**
shift right data for two bit positions.

305. The article of manufacture for fast decoding of claim **304, wherein**
said step that adds the second result **causes the machine to** subtract the second result from the input sample to produce a fourth result; and
said step that adds the first result **causes the machine to** subtract the first result from the delayed result to produce a third result.

306. The article of manufacture for fast decoding of claim **293, wherein**
said step that inverse non-stationary cell filters **further causes the machine to**:
delay an input sample for *w* samples to produce a first delayed result in each first cycle;
delay the first delayed result for *w* samples to produce a second delayed result;
delay the second delayed result for *w* samples to produce a third delayed result;
multiply the input sample with a first filter coefficient to produce a first result;
multiply the first delayed result with a second filter coefficient to produce a second result;
multiply the second delayed result with a third filter coefficient to produce a third result;
multiply the third delayed result with a fourth filter coefficient to produce a fourth result;
add the fourth result with the second result to produce a sixth result;
add the first result with the third result to produce a fifth result;
add the sixth result with the input sample to produce an eight result;
delay the eight result for *w* samples to produce the first delayed result in each second cycle;
add the fifth result with the third delayed result to produce a seventh result;
output the seventh result in each first cycle; and
output the third delayed result in each second cycle.

307. The article of manufacture for fast decoding of claim **306, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

308. The article of manufacture for fast decoding of claim **307, wherein** said operation **causes the machine to** shift right data for four bit positions.

309. The article of manufacture for fast decoding of claim **308, wherein**

said step that adds the fourth result **causes the machine to** subtract the fourth result from the second result to produce a sixth result; and

said step that adds the first result **causes the machine to** subtract the first result from the third result to produce a fifth result.

310. The article of manufacture for fast decoding of claim **280, wherein**

said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**

update adaptive histograms.

311. The article of manufacture for fast decoding of claim **310, wherein** said step that updates adaptive histograms **causes the machine to** low-pass filter probabilities selected from a group **consisting of**:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding said current symbol x.

312. The article of manufacture for fast decoding of claim **311, wherein** said step that updates adaptive histograms **further causes the machine to** adapt a dominant pole during said low-pass filtering.

313. The article of manufacture for fast decoding of claim **312, wherein** said step that adapts a dominant pole **causes the machine to** halve value of the dominant pole in each adaptation cycle.

314. The article of manufacture for fast decoding of claim **280, wherein** said step that entropy decodes is a step that range decode, **causing the machine to** multiply a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

315. The article of manufacture for fast decoding of claim **314, wherein** said step that multiplies **causes the machine to**:

simplified multiply a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shift left the first result for l bit positions.

316. The article of manufacture for fast decoding of claim **314, wherein** said step that multiplies **causes the machine to** shift left said number $Q(x)$ for l bit positions.

317. The article of manufacture for fast decoding of claim **314, wherein** said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

318. The article of manufacture for fast decoding of claim **314, wherein** said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shift left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

319. The article of manufacture for fast decoding of claim **280**, **wherein**

said step that entropy decodes is a step that range decode, **causing the machine to**

divide a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

320. The article of manufacture for fast decoding of claim **319**, **wherein**

said step that divides **causes the machine to**

shift right said range R for $w_3 = \log_2(Total)$ bit positions.

321. The article of manufacture for fast decoding of claim **280**, **wherein**

said step that entropy decodes is a step that range decode, **causing the machine to**

divide a bottom range limit B with a prescaled range r, to produce a range correction $t = \lfloor B/r \rfloor$.

322. The article of manufacture for fast decoding of claim **321**, **wherein**

said step that divides **causes the machine to:**

simplified divide said bottom range limit B with a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shift right the first result for l bit positions.

323. The article of manufacture for fast decoding of claim **321**, **wherein**

said step that divides **causes the machine to:**

multiply said bottom range limit B with a first predefined number, dependent on a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shift right the first result for a sum of l and a second predefined number of bit positions, dependent on said small number V.

324. The article of manufacture for fast decoding of claim **280, wherein**

said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**

reconstruct transformation coefficient C, using a magnitude-set index MS, a sign S and a residual R.

325. The article of manufacture for fast decoding of claim **324, wherein**

said step that entropy decodes **causes the machine to**

decode said residual R using inverse variable length coding (INVVLC).

326. The article of manufacture for fast decoding of claim **324, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

context model a transformation coefficient to be decoded, using already decoded transformation coefficients.

327. The article of manufacture for fast decoding of claim **326, wherein**

said already decoded transformation coefficients are located north-east, north, north-west and west from the transformation coefficient to be decoded.

328. The article of manufacture for fast decoding of claim **326, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a mean value $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already decoded transformation coefficients.

329. The article of manufacture for fast decoding of claim **328, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

limit a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

330. The article of manufacture for fast decoding of claim **329, wherein**

said step that entropy decodes **causes the machine to**

range decode said magnitude-set index MS as a current symbol x, using said adaptive magnitude histogram $h[MC]$.

331. The article of manufacture for fast decoding of claim **330, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update an adaptive magnitude histogram $h[MC]$ using said magnitude-set index MS as a current symbol x.

332. The article of manufacture for fast decoding of claim **326, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a ternary context TC as the ternary code of sign values S_i of said already decoded transformation coefficients.

333. The article of manufacture for fast decoding of claim **332, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

translate said ternary context TC into a sign context SC using CTX table.

334. The article of manufacture for fast decoding of claim **333, wherein**

said step that entropy decodes **causes the machine to**

range decode sign S as a current symbol x, using an adaptive sign histogram $g[SC]$.

335. The article of manufacture for fast decoding of claim **334, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update an adaptive sign histogram $g[SC]$ using decoded sign S as a current symbol x.

336. The article of manufacture for fast decoding of claim **334, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

invert less probable decoded sign S using NEG table.

337. A data signal for fast encoding of input data into output compressed data embodied in a carrier wave **comprising** a machine readable code **causing the machine to**:

direct subband transform the input data to produce transformation coefficients;

estimate the probabilities of symbols within the specified contexts using the transformation coefficients to produce the probabilities of symbols within the specified contexts;

entropy encode transformation coefficients using the probabilities of symbols within the specified contexts to produce encoded data; and

substantially synchronize the encoded data to produce the output compressed data,

whereby machine performs lossless compression.

338. The data signal for fast encoding of claim **337, further causing the machine to**

quantize the transformation coefficients to produce quantized transformation coefficients, **wherein**:

said step that estimates the probabilities of symbols within the specified contexts is performed using the quantized transformation coefficients to produce the probabilities of symbols within the specified contexts; and

said entropy encode quantized transformation coefficients is performed using the probabilities of symbols within the specified contexts to produce encoded data,

whereby machine performs lossy compression.

339. The data signal for fast encoding of claim **337**, **further causing the machine to**

substantially synchronize the encoded data to produce synchronized compressed data, **wherein**

buffering the synchronized compressed data is performed to produce the output compressed data.

340. The data signal for fast encoding of claim **337**, **further causing the machine to**

color space convert original input data to produce input data.

341. The data signal for fast encoding of claim **337**, **wherein**

said step that direct subband transforms **causes the machine to**:

(a) direct subband transform the input data to produce transformation coefficients;

(b) direct subband transform selected transformation coefficients to produce transformed transformation coefficients; and

(c) repeat step (b) finite number of times.

342. The data signal for fast encoding of claim **341**, **wherein** said selected transformation coefficients are low-pass transformed for one-dimensional input data.

343. The data signal for fast encoding of claim **341**, **wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

344. The data signal for fast encoding of claim **337**, **wherein**

said step that direct subband transforms **causes the machine to**:

direct filter horizontally; and

direct filter vertically.

345. The data signal for fast encoding of claim **344**, **wherein** said step that direct filters horizontally is different from said step that direct filters vertically.

346. The data signal for fast encoding of claim **344, wherein**

at least one of said step that direct filters horizontally and said step that direct filters vertically **causes the machine to**

direct non-stationary filter.

347. The data signal for fast encoding of claim **337, wherein**

said step that direct subband transforms **causes the machine to**

direct filter.

348. The data signal for fast encoding of claim **347, wherein**

said step that direct filters **causes the machine to**

direct non-stationary filter.

349. The data signal for fast encoding of claim **348, wherein**

said step that direct non-stationary filters **causes the machine to**

perform a plurality of successive direct non-stationary cell filterings.

350. The data signal for fast encoding of claim **349, wherein**

said step that direct non-stationary cell filters **causes the machine to:**

filter using first direct transfer function in the first cycle; and

filter using second direct transfer function in the second cycle.

351. The data signal for fast encoding of claim **350, wherein**

said first cycle is active during horizontal filtering of each second pixel; and

said second cycle is active during horizontal filtering of other pixels.

352. The data signal for fast encoding of claim **350, wherein**

said first cycle is active during vertical filtering of each second line; and

said second cycle is active during vertical filtering of other lines.

353. The data signal for fast encoding of claim **350, wherein**

said step that direct non-stationary filters **further causes the machine to:**

multiply a result of a plurality of steps that successive direct non-stationary cell filter with a first gain number to produce a first result;

multiply a result of a plurality of steps that successive direct non-stationary cell filter with a second gain number to produce a second result;

select the first result in each first cycle to produce an output sample; and

select the second result in each second cycle to produce the output sample.

354. The data signal for fast encoding of claim **350, wherein**

said step that direct non-stationary cell filters **further causes the machine to**:

delay an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;

multiply each even indexed delayed result with appropriate first filter coefficient selected from a group of first filter coefficients to produce first results;

add together all first results to produce a third result;

add the third result with the input sample to produce a fifth result;

delay the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;

multiply the input sample and each odd indexed delayed result with appropriate second filter coefficient selected from a group of second filter coefficients to produce second results;

add together all second results to produce a fourth result;

add the fourth result with last delayed result to produce a sixth result;

output the sixth result in each first cycle; and

output the last delayed result in each second cycle.

355. The data signal for fast encoding of claim **354, wherein**

at least one said step that multiplies **causes the machine to**

perform an operation selected from a group **consisting of**:

shift and bit remap.

356. The data signal for fast encoding of claim **354, wherein**

said step that direct non-stationary filters **further causes the machine to**:

transform the third result by first function N_1 to produce a seventh result;

transform the fourth result by first function N_2 to produce an eight result;
add the seventh result with the input sample to produce a fifth result; and
add the eight result with last delayed result to produce a sixth result.

357. The data signal for fast encoding of claim **356, wherein** at least one said step that transforms **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

358. The data signal for fast encoding of claim **350, wherein** said step that direct non-stationary cell filters **further causes the machine to**:
delay an input sample for w samples to produce a delayed result in each first cycle;
multiply the delayed result with a second filter coefficient to produce a second result;
add the second result with the input sample to produce a fourth result;
delay the fourth result for w samples to produce the delayed result in each second cycle;
multiply the input sample with a first filter coefficient to produce a first result;
add the first result with the delayed result to produce a third result;
output the third result in each first cycle; and
output the delayed result in each second cycle.

359. The data signal for fast encoding of claim **358, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

360. The data signal for fast encoding of claim **359, wherein** said operation **causes the machine to** shift right data for two bit positions.

361. The data signal for fast encoding of claim **359, wherein** said operation **causes the machine to** shift right data for one bit position.

362. The data signal for fast encoding of claim **361, wherein**

said step that adds the second result **causes the machine to** subtract the second result from the input sample to produce a fourth result; and

said step that adds the first result **causes the machine to** subtract the first result from the delayed result to produce a third result.

363. The data signal for fast encoding of claim **350, wherein**

said step that direct non-stationary cell filters **further causes the machine to**:

delay the input sample for w samples to produce a first delayed result in each first cycle;

delay the first delayed result for w samples to produce a second delayed result;

delay the second delayed result for w samples to produce a third delayed result;

multiply the input sample with a first filter coefficient to produce a first result;

multiply the first delayed result with a second filter coefficient to produce a second result;

multiply the second delayed result with a third filter coefficient to produce a third result;

multiply the third delayed result with a fourth filter coefficient to produce a fourth result;

add the second result with the fourth result to produce a sixth result;

add the third result with the first result to produce a fifth result;

add the sixth result with the input sample to produce an eight result;

delay the eight result for w samples to produce the first delayed result in each second cycle;

add the fifth result with the third delayed result to produce a seventh result;

output the seventh result in each first cycle; and

output the third delayed result in each second cycle.

364. The data signal for fast encoding of claim **363, wherein**

at least one said step that multiplies **causes the machine to**

perform an operation selected from a group **consisting of**:

shift and bit remap.

365. The data signal for fast encoding of claim **364, wherein**

said operation **causes the machine to**

shift right data for four bit positions.

366. The data signal for fast encoding of claim **365, wherein**:

said step that adds the second result **causes the machine to** subtract the second result from the fourth result to produce a sixth result; and

said step that adds the third result **causes the machine to** subtract the third result from the first result to produce a fifth result.

367. The data signal for fast encoding of claim **337, wherein**

said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**

update adaptive histograms.

368. The data signal for fast encoding of claim **367, wherein**

said step that updates adaptive histograms **causes the machine to**

low-pass filter probabilities selected from a group **consisting of**:

probabilities of occurrences of a current symbol x; and

cumulative probabilities of occurrences of all symbols preceding the current symbol x.

369. The data signal for fast encoding of claim **368, wherein**

said step that updates adaptive histograms **further causes the machine to**

adapt a dominant pole during said low-pass filtering.

370. The data signal for fast encoding of claim **369, wherein**

said step that adapts a dominant pole **causes the machine to**

halve value of the dominant pole in each adaptation cycle.

371. The data signal for fast encoding of claim **337, wherein**

said step that entropy encodes is a step that range encode, **causing the machine to**

multiply a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

372. The data signal for fast encoding of claim **371, wherein** said step that multiplies **causes the machine to**:

simplified multiply a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shift left the first result for l bit positions.

373. The data signal for fast encoding of claim **371, wherein** said step that multiplies **causes the machine to**

shift left said number $Q(x)$ for l bit positions.

374. The data signal for fast encoding of claim **371, wherein** said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

375. The data signal for fast encoding of claim **371, wherein** said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shift left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

376. The data signal for fast encoding of claim **337, wherein** said step that entropy encodes is a step that range encode, **causing the machine to**:

divide a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

377. The data signal for fast encoding of claim **376, wherein** said step that divides **causes the machine to** shift right range R for $w_3 = \log_2(Total)$ bit positions.

378. The data signal for fast encoding of claim **337, wherein** said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**:

split a transformation coefficient C into a sign S and a magnitude M;

determine a magnitude-set index MS using said magnitude M and a magnitude-set table; and

determine a residual R using said magnitude M and said magnitude-set table.

379. The data signal for fast encoding of claim **378, wherein**

said magnitude-set index MS is determined to be equal to a sum of a doubled position of the first nonzero bit of the highest significance and the value of the first next bit of the lower significance in a binary representation of said magnitude M; and

said residual R is determined as the difference between said magnitude M and the lower coefficient limit, equal to a value of said magnitude M with all bits zeroed except the first nonzero bit of the highest significance and the first next bit of the lower significance in a binary representation of said magnitude M.

380. The data signal for fast encoding of claim **378, wherein** said step that entropy encodes **causes the machine to** encode a residual R using variable length coding (VLC).

381. The data signal for fast encoding of claim **378, wherein** said estimates the probabilities of symbols within the specified contexts **further causes the machine to**

context model a transformation coefficient to be encoded, using already encoded transformation coefficients.

382. The data signal for fast encoding of claim **381, wherein**

said already encoded transformation coefficients are located north-east, north, north-west and west from said transformation coefficient to be encoded.

383. The data signal for fast encoding of claim **381, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a mean value $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already encoded transformation coefficients.

384. The data signal for fast encoding of claim **383, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

limit a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

385. The data signal for fast encoding of claim **384, wherein**

said step that entropy encodes **causes the machine to**

range encode said magnitude-set index MS as a current symbol x, using said adaptive magnitude histogram $h[MC]$.

386. The data signal for fast encoding of claim **384, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update an adaptive magnitude histogram $h[MC]$ using said magnitude-set index MS as a current symbol x.

387. The data signal for fast encoding of claim **381, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a ternary context TC as the ternary code of sign values S_i of said already encoded transformation coefficients.

388. The data signal for fast encoding of claim **387, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

invert less probable said sign S using NEG table.

389. The data signal for fast encoding of claim **387, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

translate said ternary context TC into a sign context SC using CTX table.

390. The data signal for fast encoding of claim **389, wherein**

said step that entropy encodes **causes the machine to**

range encode said sign S as a current symbol x, using said adaptive sign histogram $g[SC]$.

391. The data signal for fast encoding of claim **389, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update of an adaptive sign histogram $g[SC]$ using said sign S as a current symbol x.

392. A data signal for fast decoding of input compressed data into output data embodied in a carrier wave **comprising** a machine readable code **causing the machine to**:

substantially synchronize the input compressed data to produce synchronized compressed data;

entropy decode the synchronized compressed data using the probabilities of symbols within the specified contexts to produce transformation coefficients;

estimate the probabilities of symbols within the specified contexts using the transformation coefficients to produce the probabilities of symbols within the specified contexts; and

inverse subband transform the transformation coefficients to produce the output data, **whereby** machine performs lossless decompression.

393. The data signal for fast decoding of claim **392, further causing the machine to**

dequantize transformation coefficients to produce dequantized transformation coefficients,

wherein

inverse subband transforming the dequantized transformation coefficients is performed to produce the output data,

whereby machine performs lossy decompression.

394. The data signal for fast decoding of claim **392, further causing the machine to**

buffer the input compressed data to produce buffered compressed data,

wherein

substantially synchronizing the buffered compressed data is performed to produce the synchronized compressed data.

395. The data signal for fast decoding of claim **392, further causing the machine to**

color space convert the output data to produce converted output data.

396. The data signal for fast decoding of claim **392, wherein**

said step that inverse subband transforms **causes the machine to**:

(a) inverse subband transform transformation coefficients to produce selected transformation coefficients;

(b) repeat step (a) finite number of times; and

(c) inverse subband transform transformation coefficients to produce the output data.

397. The data signal for fast decoding of claim **396, wherein** said selected transformation coefficients are low-pass transformed for one-dimensional input data.

398. The data signal for fast decoding of claim **396, wherein** said selected transformation coefficients are low-pass transformed both horizontally and vertically for two-dimensional input data.

399. The data signal for fast decoding of claim **392, wherein** said step that inverse subband transforms **causes the machine to**: inverse filter horizontally; and inverse filter vertically.

400. The data signal for fast decoding of claim **399, wherein** said step that inverse filters horizontally is different from said step that inverse filters vertically.

401. The data signal for fast decoding of claim **399, wherein** at least one of said step that inverse filters horizontally and said step that inverse filters vertically **causes the machine to** inverse non-stationary filter.

402. The data signal for fast decoding of claim **392, wherein** said step that inverse subband transforms **causes the machine to** inverse filter.

403. The data signal for fast decoding of claim **402, wherein** said step that inverse filters **causes the machine to** inverse non-stationary filter.

404. The data signal for fast decoding of claim **403, wherein** said step that inverse non-stationary filters **causes the machine to** perform a plurality of successive inverse non-stationary cell filterings.

405. The data signal for fast decoding of claim **404, wherein** said step that inverse non-stationary cell filters **causes the machine to**: filter using first direct transfer function in the first cycle; and filter using second direct transfer function in the second cycle.

406. The data signal for fast decoding of claim **405, wherein**
said second cycle is active during horizontal filtering of each second pixel; and
said first cycle is active during horizontal filtering of other pixels.

407. The data signal for fast decoding of claim **405, wherein**
said second cycle is active during vertical filtering of each second line; and
said first cycle is active during vertical filtering of other lines.

408. The data signal for fast decoding of claim **405, wherein**
said step that inverse non-stationary filters **further causes the machine to**:
multiply an input with a reciprocal value of a first gain number to produce a first result;
multiply an input with a reciprocal value of a second gain number to produce a second result;
select the first result in each second cycle to produce an input sample for a plurality of steps that successive inverse non-stationary cell filter; and
select the second result in each first cycle to produce an input sample for a plurality of steps that successive inverse non-stationary cell filter.

409. The data signal for fast decoding of claim **405, wherein**
said step that inverse non-stationary cell filters **further causes the machine to**:
delay an input sample for w samples to produce a plurality of even and odd indexed delayed results in each first cycle;
multiply each even indexed delayed result with appropriate first filter coefficient selected from a group of first filter coefficients to produce first results;
add together all first results to produce a third result;
add the third result with the input sample to produce a fifth result;
delay the fifth result for w samples to produce a plurality of even and odd indexed delayed results in each second cycle;
multiply the input sample and each odd indexed delayed result with appropriate second filter coefficient selected from a group of second filter coefficients to produce second results;
add together all second results to produce a fourth result;
add the fourth result with last delayed result to produce a sixth result;

output the sixth result in each first cycle; and
output the last delayed result in each second cycle.

410. The data signal for fast decoding of claim **409, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

411. The data signal for fast decoding of claim **409, wherein** said step that inverse non-stationary cell filters **further causes the machine to**
transform the third result by a first function N_1 to produce a seventh result;
transform the fourth result by a first function N_2 to produce an eight result;
add the seventh result with the input sample to produce a fifth result; and
add the eight result with last delayed result to produce a sixth result.

412. The data signal for fast decoding of claim **411, wherein** at least one said step that transforms **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

413. The data signal for fast decoding of claim **405, wherein** said step that inverse non-stationary cell filters **further causes the machine to**:
delay an input sample for w samples to produce a delayed result in each first cycle;
multiply the delayed result with a second filter coefficient to produce a second result;
add the second result with the input sample to produce a fourth result;
delay the fourth result for w samples to produce the delayed result in each second cycle;
multiply the input sample with a first filter coefficient to produce a first result;
add the first result with the delayed result to produce a third result;
output the third result in each first cycle; and
output the delayed result in each second cycle.

414. The data signal for fast decoding of claim **413, wherein** at least one said step that multiplies **causes the machine to** perform an operation selected from a group **consisting of**: shift and bit remap.

415. The data signal for fast decoding of claim **414, wherein** said operation **causes the machine to** shift right for one bit position.

416. The data signal for fast decoding of claim **414, wherein** said operation **causes the machine to** shift right data for two bit positions.

417. The data signal for fast decoding of claim **416, wherein**

said step that adds the second result **causes the machine to** subtract the second result from the input sample to produce a fourth result; and

said step that adds the first result **causes the machine to** subtract the first result from the delayed result to produce a third result.

418. The data signal for fast decoding of claim **405, wherein**

said step that inverse non-stationary cell filters **further causes the machine to**:

delay an input sample for w samples to produce a first delayed result in each first cycle;

delay the first delayed result for w samples to produce a second delayed result;

delay the second delayed result for w samples to produce a third delayed result;

multiply the input sample with a first filter coefficient to produce a first result;

multiply the first delayed result with a second filter coefficient to produce a second result;

multiply the second delayed result with a third filter coefficient to produce a third result;

multiply the third delayed result with a fourth filter coefficient to produce a fourth result;

add the fourth result with the second result to produce a sixth result;

add the first result with the third result to produce a fifth result;

add the sixth result with the input sample to produce an eight result;

delay the eight result for w samples to produce the first delayed result in each second cycle;

add the fifth result with the third delayed result to produce a seventh result;
output the seventh result in each first cycle; and
output the third delayed result in each second cycle.

419. The data signal for fast decoding of claim **418, wherein**
at least one said step that multiplies **causes the machine to**
perform an operation selected from a group **consisting of**:
shift and bit remap.

420. The data signal for fast decoding of claim **419, wherein**
said operation **causes the machine to**
shift right data for four bit positions.

421. The data signal for fast decoding of claim **420, wherein**
said step that adds the fourth result **causes the machine to** subtract the fourth result from the second result to produce a sixth result; and
said step that adds the first result **causes the machine to** subtract the first result from the third result to produce a fifth result.

422. The data signal for fast decoding of claim **392, wherein**
said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**
update adaptive histograms.

423. The data signal for fast decoding of claim **422, wherein**
said step that updates adaptive histograms **causes the machine to**
low-pass filter probabilities selected from a group **consisting of**:
probabilities of occurrences of a current symbol x; and
cumulative probabilities of occurrences of all symbols preceding said current symbol x.

424. The data signal for fast decoding of claim **423, wherein** said step that updates adaptive histograms **further causes the machine to** adapt a dominant pole during said low-pass filtering.

425. The data signal for fast decoding of claim **424, wherein** said step that adapts a dominant pole **causes the machine to** halve value of the dominant pole in each adaptation cycle.

426. The data signal for fast decoding of claim **392, wherein** said step that entropy decodes is a step that range decode, **causing the machine to** multiply a prescaled range r with a number $Q(x)$ selected from a group **consisting of**:

a number $U(x)$ of occurrences of all symbols preceding a current symbol x, to produce a range correction $t = r \cdot U(x)$; and

a number $u(x)$ of occurrences of the current symbol x, to produce a range $R = r \cdot u(x)$.

427. The data signal for fast decoding of claim **426, wherein** said step that multiplies **causes the machine to**:

simplified multiply a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ with said number $Q(x)$ to produce a first result; and

shift left the first result for l bit positions.

428. The data signal for fast decoding of claim **426, wherein** said step that multiplies **causes the machine to** shift left said number $Q(x)$ for l bit positions.

429. The data signal for fast decoding of claim **426, wherein** said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is any odd number higher or equal 3;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

430. The data signal for fast decoding of claim **426, wherein**

said step that multiplies **causes the machine to**:

zero a first result, when a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ is equal 1;

shift left said number $Q(x)$ for one bit position to produce a first result, when said small number V is equal 3;

shift left said number $Q(x)$ for two bit positions to produce a first result, when said small number V is any odd number higher or equal 5;

add said number $Q(x)$ with the first result to produce a second result; and

shift left the second result for l bit positions.

431. The data signal for fast decoding of claim **392, wherein**

said step that entropy decodes is a step that range decode, **causing the machine to**

divide a range R with a number *Total* of occurrences of all symbols, to produce a prescaled range $r = \lfloor R/Total \rfloor$.

432. The data signal for fast decoding of claim **431, wherein**

said step that divides **causes the machine to**

shift right said range R for $w_3 = \log_2(Total)$ bit positions.

433. The data signal for fast decoding of claim **392, wherein**

said step that entropy decodes is a step that range decode, **causing the machine to**

divide a bottom range limit B with a prescaled range r, to produce a range correction $t = \lfloor B/r \rfloor$.

434. The data signal for fast decoding of claim **433, wherein**

said step that divides **causes the machine to**:

simplified divide said bottom range limit B with a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shift right the first result for l bit positions.

435. The data signal for fast decoding of claim **433, wherein**

said step that divides **causes the machine to:**

multiply said bottom range limit B with a first predefined number, dependent on a small number $V = \lfloor r \cdot 2^{-l} \rfloor$ to produce a first result; and

shift right the first result for a sum of l and a second predefined number of bit positions, dependent on said small number V.

436. The data signal for fast decoding of claim **392, wherein**

said step that estimates the probabilities of symbols within the specified contexts **causes the machine to**

reconstruct transformation coefficient C, using a magnitude-set index MS, a sign S and a residual R.

437. The data signal for fast decoding of claim **436, wherein**

said step that entropy decodes **causes the machine to**

decode said residual R using inverse variable length coding (INVVLC).

438. The data signal for fast decoding of claim **436, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

context model a transformation coefficient to be decoded, using already decoded transformation coefficients.

439. The data signal for fast decoding of claim **438, wherein**

said already decoded transformation coefficients are located north-east, north, north-west and west from the transformation coefficient to be decoded.

440. The data signal for fast decoding of claim **438, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a mean value $\overline{MS}$ as the mean value of magnitude-set indexes MS_i of said already decoded transformation coefficients.

441. The data signal for fast decoding of claim **440, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

limit a maximum mean value $\overline{MS}$ by a constant ML to produce a magnitude context MC.

442. The data signal for fast decoding of claim **441, wherein**

said step that entropy decodes **causes the machine to**

range decode said magnitude-set index MS as a current symbol x, using said adaptive magnitude histogram $h[MC]$.

443. The fast decoding of claim **442, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update an adaptive magnitude histogram $h[MC]$ using said magnitude-set index MS as a current symbol x.

444. The data signal for fast decoding of claim **438, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

determine a ternary context TC as the ternary code of sign values S_i of said already decoded transformation coefficients.

445. The data signal for fast decoding of claim **444, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

translate said ternary context *TC* into a sign context *SC* using CTX table.

446. The data signal for fast decoding of claim **445, wherein**

said step that entropy decodes **causes the machine to**

range decode sign *S* as a current symbol *x*, using an adaptive sign histogram $g[SC]$.

447. The data signal for fast decoding of claim **446, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

update an adaptive sign histogram $g[SC]$ using decoded sign *S* as a current symbol *x*.

448. The data signal for fast decoding of claim **446, wherein**

said step that estimates the probabilities of symbols within the specified contexts **further causes the machine to**

invert less probable decoded sign *S* using NEG table.

449. An article of manufacture for fast encoding of input data into output compressed data **comprising** a storage medium with a machine readable code **causing the machine to** perform the method of claim **113.**

450. An article of manufacture for fast decoding of input compressed data into output data **comprising** a storage medium with a machine readable code **causing the machine to** perform the method of claim **168.**

451. A data signal for fast encoding of input data into output compressed data embodied in a carrier wave **comprising** a machine readable code **causing the machine to** perform the method of claim **113.**

452. A data signal for fast decoding of input compressed data into output data embodied in a carrier wave **comprising** a machine readable code **causing the machine to** perform the method of claim **168.**